830-H-10

NAS1.55:3059

APR 2 0 1990

NASA Conference Publication 3059

Third Annual Workshop on Space Operations Automation and Robotics (SOAR '89)

Proceedings of a workshop hosted by
Lyndon B. Johnson Space Center
Houston, Texas
July 25-27, 1989

NASA

NASA Conference Publication 3059

Third Annual Workshop on Space Operations Automation and Robotics (SOAR '89)

Sandy Griffin, *Editor*
NASA Lyndon B. Johnson Space Center
Houston, Texas

Proceedings of a workshop sponsored by the National Aeronautics and Space Administration, Washington, D.C., the U.S. Air Force, Washington, D.C., and cosponsored by the University of Houston-Clear Lake, Houston, Texas, and hosted by Lyndon B. Johnson Space Center
Houston, Texas
July 25-27, 1989



National Aeronautics and Space Administration

Office of Management

Scientific and Technical Information Division

1990

PREFACE

This document represents the proceedings of the Third Annual Workshop on Space Operations Automation and Robotics, otherwise known as SOAR '89, which was held at the Lyndon B. Johnson Space Center on July 25-27, 1989, in Houston, Texas.

This workshop was jointly sponsored by the National Aeronautics and Space Administration and the United States Air Force. It was cosponsored by the University of Houston - Clear Lake. SOAR '89 helped to establish communications between individuals and organizations involved in similar research and technology. It brought together project/program managers in open exchange through presentation of technical papers and panel discussions. The objective of SOAR '89 was to provide a vehicle for engineers, scientists, and managers of both the Air Force and NASA to come together in a workshop environment and exchange ideas, problems/problem solutions, and technical information on projects of mutual interest and, perhaps most importantly, to build a solid foundation for future interaction and cooperation. The workshop consisted of technical sessions emphasizing AI/Expert Systems, Human Factors, Environment, Robotics, and Application Development and Transition. The workshop will rotate annually between NASA and an Air Force installation.

The papers included in these proceedings were published in general as received from the authors with minimum modification and editing. Information contained in the individual papers is not to be construed as being officially endorsed by NASA.

MESSAGE FROM THE GENERAL CHAIR AND ASSISTANT GENERAL CHAIR

The previous SOAR workshops (1987 and 1988) have set a standard of excellence I am confident will be met by SOAR'89. These gatherings of NASA and Air Force experts in automation and robotics have produced significant results in two ways: duplication of effort has been reduced and new application areas for existing technology have been revealed.

As the Nation's space program gathers momentum, the Air Force and NASA face a number of technical challenges. In addition to designing and constructing Space Station Freedom, both agencies are actively cooperating in the development of a new generation of expendable launch vehicles. Both programs offer a superb opportunity for the infusion of automation and robotics technology into ground- and space-based systems.

We look forward to SOAR'89 strengthening and extending the links forged between NASA and the Air Force in the previous workshops. Cooperative programs such as this benefit not only our two agencies, but serve as an exemplary model for other government, commercial, and academic entities.

I deeply appreciate the continued and deepening Air Force support for SOAR and look forward to an exciting and profitable interchange of ideas.

Robert H. Brown
NASA/Johnson Space Center

The SOAR'89 workshop will bring together Air Force and NASA project/program managers and members of the technical community for an information exchange on space operations. The visibility that the SOAR '89 workshop will provide attendees into technology applications will give us a singular opportunity to establish additional cooperative technology development and transition programs.

In addition to automation and robotics, the SOAR'89 will also include sessions on Space Debris and Technology Insertion. Each session will start with Air Force and NASA programmatic overviews of present efforts which will be followed by technical papers and conclude with panel discussions of problems/solutions within the topic area.

With your participation, we look forward to an informative and productive workshop.

Col. Paul C. Anderson
USAF/Space Technology Center

ACKNOWLEDGEMENT

Acknowledgements are due to all personnel who provided the logistic support necessary to the success of this workshop. Thanks are also due to Computer Science Corporation, Lincom, Barrios Technology, McDonnell-Douglas Space Operations Corporation, and Omniplan Corporation in extending the support of their personnel.

THIRD ANNUAL WORKSHOP ON SPACE OPERATIONS AUTOMATION AND ROBOTICS

SOAR '89 Organizing Committee

General Chair	Robert H. Brown	NASA/JSC
Assistant General Chair	Col. Paul C. Anderson	AFSTC/USAF
Technical Chair	Robert T. Savely	NASA/JSC
Tutorial Chair	Chris Culbert	NASA/JSC
Executive Chair	Sandy Griffin	NASA/JSC
Administrative Co-Chairs	Glenn Freedman	University of Houston-Clear Lake
	Carla Armstrong	Barrios Technology, Inc.
Exhibit Chair	Ellis Henry	NASA/JSC
Registration and Publicity	Katherine Moser	University of Houston-Clear Lake

Technical Area Developers

Robotics	Jack Pennington	NASA/Langley Research Center
	Capt. Ron Julian	AAMRL/BBA/USAF
Intelligent Systems	Gregg Swietek	NASA/Headquarters
	Dave Weeks	NASA/Marshall Space Flight Center
	Lt. Mike Wellman	AFWAL/TXI/USAF
Environment	Carolyn Purvis	NASA/Lewis Research Center
	Charley Pike	AFGL/PHK/USAF
Human Factors	Michael Shafto	NASA/Ames Research Center
	Bob Bachert	AAMRL/HID/USAF

CONTENTS

KEYNOTE SPEECH TO SOAR '89 CONFERENCE

James B. Odom
Applied Research, Inc. 1

SESSION 1: **Air Force/NASA Artificial Intelligence Overview Panel**
SESSION CHAIR: **Brigadier General J. Van Blois, USAF**
PANEL MEMBERS: **Captain Mark A. Gersh, AFSC, Andrews AFB**
Dr. P. Friedland, NASA/Ames Research Center
G. Swietek, NASA Headquarters

Artificial Intelligence Within the Air Force Systems Command (AFSC)
Captain Mark A. Gersh 13

SESSION 2: **Space-Based Manufacturing and Assembly**
SESSION CHAIR: **Dr. Mike Walker, University of Michigan**

Preliminary Tests of Automated Assembly of Space Structures
(Paper not provided by publication date) 47

Experimental Results of Integrated EVA/Telerobotic Work Sites
(Paper not provided by publication date) 49

Mobile Transporter Path Planning 51

SESSION 3: **Human Factors in Hazardous Environments**
SESSION CO-CHAIRS: **Captain Al Reinhardt, AFOSR and NASA/Ames Research Center**
Dr. R. Drawbaugh, AFHSD, Wright-Patterson AFB

Space Station Space Suit Test Program 61

AX-5 Advanced Space Suit Design Overview
(Paper not provided by publication date) 91

Combat Edge: Advances in Tactical Fighter Pressure Garments
(Paper not provided by publication date) 93

Multisystem Approach to Manned Diving Operation
(Paper not provided by publication date) 95

SESSION 4: **Spacecraft Contamination**
SESSION CHAIR: **C. Pike, Geophysics Laboratory, Hanscom AFB**

Spacecraft Contamination Programs within the Air Force Systems Command Laboratories 97

Contamination of Spacecraft by Recontact of Dumped Liquids 99

Spacecraft Glow Current Status (Paper not provided by publication date) 105

SESSION 5: **Monitoring and Diagnosis**
SESSION CO-CHAIRS: **Dave Weeks, NASA/Marshall Space Flight Center**
Captain Mark Gersh, Andrews AFB

Automation in the Space Station Module Power Management and Distribution Breadboard 107

NASA Systems Autonomy Demonstration Project: Advanced Automation Demonstration of Space Station Freedom Thermal Control System 113

StarPlan: A Model-Based Diagnostic System for Spacecraft 117

Smart Built-In Test 123

Solar Array Automation Limitations 127

SESSION 6: **Programs and Future Directions**
SESSION CHAIR: **Jack Pennington, NASA/Langley Research Center**

A Perspective on Robotic Research Programs (Paper not provided by publication date) 131

SDIO Robotics in Space Applications 133

SESSION 7: **Anthropometric Computer Models**
SESSION CO-CHAIRS: **Dr. Betty Goldsberry, Lockheed Engineering and Sciences Company**
Dr. Joe McDaniel, AAMRL/HEG, Wright-Patterson AFB

CREW CHIEF Animation Techniques: Data Applications from a Space Suit Study (Paper not provided by publication date) 137

CREW CHIEF: A Computer Graphics Simulation of an Aircraft Maintenance Technician 139

Modeling Strength Data for CREW CHIEF 143

The Use of PLAID in the Space Station Freedom Viewing Analysis 149

Quantitative Assessment of Human Motion Using Video Motion Analysis 155

SESSION 8: Spacecraft Contamination
SESSION CHAIR: S. Mende, Lockheed Missiles and Space Company

Gaseous Optical Contamination of the Spacecraft Environment: A Review 159

Contam 3.4 (SDI Applications Version): Status Report (Paper not provided by publication date) 167

The Altitude and Angular Dependence of Radiation Contamination (Paper not provided by publication date) 169

Spacecraft External Molecular Contamination Analysis 171

An Assessment of Propulsion System Contamination on Space Platform Operations (Paper not provided by publication date) 179

SESSION 9: Monitoring and Control
SESSION CO-CHAIRS: Dave Weeks, NASA/Marshall Space Flight Center
Captain Mark Gersh, Andrews AFB

Autonomous Power Expert Fault Diagnostic System for Space Station Freedom Electrical Power System Testbed 181

An Expert System to Advise Astronauts During Experiments: The Protocol Manager Module 187

The Environmental Control and Life Support System Advanced Automation Project 195

Monitoring and Control of Spacecraft Systems Using Procedural Reasoning 209

Knowledge Representation to Support Reasoning Based on Multiple Models 219

SESSION 10: Telepresence: State of the Concept and the Technology
SESSION CHAIR: Dr. Betty Goldsberry, Lockheed Engineering and Sciences Company

Telepresence for Space: The State of the Concept 223

Telepresence and Space Station Freedom Workstation Operations 229

The Human Factors of Workstation Telepresence 235

SESSION 11: **NASA and USAF Programmatic Overviews**
SESSION CO-CHAIRS: **Mike Shafto, NASA/Ames Research Center**
Bob Bachert, Wright-Patterson AFB

Overview of NASA's Technology Development for Humans in Space
(Paper not provided by publication date) 251

USAF Space Human Factors Overview
(Paper not provided by publication date) 253

SESSION 12: **Spacecraft Contamination**
SESSION CHAIR: **E. Murad, Geophysical Laboratory, Hanscom AFB**

The Particulate Environment Surrounding Space Structures
(Paper not provided by publication date) 255

Satellite Material Contaminant Optical Properties 257

Experimental Investigations of Low Energy (4-40 eV) Collisions of $O^-(^2P)$ Ions and $O(^3P)$ Atoms with Surfaces 263

Laboratory Investigations: Low Earth Orbit Environment Chemistry with Spacecraft Surfaces 269

Atomic Oxygen Effects on Spacecraft Surfaces: Orbital Measurements and Ground Simulation
(Paper not provided by publication date) 277

SESSION 13: **Mission Operations**
SESSION CHAIR: **John Muratore, NASA/Johnson Space Center**

RTDS: Real-Time Expert Systems in Shuttle Mission Control
(Paper not provided by publication date) 279

Artificial Intelligence for Multi-Mission Planetary Operations 281

CLEAR: Automating Control Centers with Expert System Technology 289

Process and Information Integration via Hypermedia 295

An Approach to a Real-Time Distribution System 299

A Comparison of Two Neural Network Schemes for Navigation 305

SESSION 14: Robotic Servicing
SESSION CHAIR: Chuck Woolley, NASA/Johnson Space Center

The Flight Telerobotic Servicer (FTS): NASA's First Operational Robotic System 311

Satellite Servicing
(Paper not provided by publication date) 319

The Telerobot Testbed: An Architecture for Remote Sensing 321

SESSION 15: Strategic Behavior and Workload
SESSION CO-CHAIRS: Sandra Hart, NASA/Ames Research Center
Dr. Michael Vidulich, AAMRL/HEG, Wright-Patterson AFB

Performance-Based Workload Assessment: Allocation Strategy and Added Task Sensitivity 329

The Workload of Sustained Attention
(Paper not provided by publication date) 337

Real-Time Measurement of Mental Workload: A Feasibility Study 339

SHAPA: An Interactive Software Tool for Protocol Analysis Applied to Aircrew Communications and Workload 347

Usefulness of Heart Measures in Flight Simulation 353

SESSION 16: Spacecraft Interaction Effects
SESSION CHAIR: D. Ferguson, NASA/Lewis Research Center

Space Environments and Their Effects on Space Automation and Robotics 361

Environmental Caused Spacecraft Anomalies in Orbit
(Paper not provided by publication date) 381

Discharge Transient Coupling in Large Space Power Systems 383

Plasma Interactions and Effects for Large Systems 393

SESSION 17: Hierarchical Control and Autonomy
SESSION CHAIR: Major Carl Lizza, WRDC/KTS, Wright-Patterson AFB

Pilot's Associate
(Paper not provided by publication date) 405

Cooperating Intelligent Agents
(Paper not provided by publication date) 407

A Task Control Architecture for Autonomous Robots 409

Automated Control of Hierarchical Systems Using Value-Driven Methods 415

Planning for Execution Monitoring on a Planetary Rover 423

SESSION 18: End Effectors/Manipulators I
SESSION CHAIR: Dr. Timothy F. Cleghorn, NASA/Johnson Space Center

Impact of End Effector Technology on Telemanipulation Performance 429

Smart Hands for the EVA Retriever 441

Kinematically Redundant Arm Formulations for Coordinated Multiple
Arm Implementations 447

Numerical Approach of Collision Avoidance and Optimal Control on
Robotic Manipulators 455

SESSION 19: Human Factors Engineering and Robotics
SESSION CO-CHAIRS: Dr. Steven R. Ellis, NASA/Ames Research Center
Dr. Thomas Malone, Carlow Associates, Inc.

Visualization of Three Dimensional Data 465

Remote Operation of an Orbital Maneuvering Vehicle in Simulated
Docking Maneuvers 471

A Helmet Mounted Display to Adapt the Telerobotic Environment
to Human Vision 477

Assessment of Control Stability for a Dexterous Teleoperator
With Time Delay
(Paper not provided by publication date) 483

Human Factors Issues in Telerobotic Systems for Space Station
Freedom Servicing 485

SESSION 20: **Spacecraft Interaction Effects**
SESSION CHAIR: **D. Ferguson, NASA/Lewis Research Center**

Interactions Between the Space Station and the Environment:
A Preliminary Assessment of EMI 493

Solar Array Arcing in Plasmas 509

Photovoltaic Array Space Power Plus Diagnostics Experiment 515

Orbital Debris and Meteoroid Effects on Spacecraft Systems
(Paper not provided by publication date) 521

SESSION 21: **Machine Learning**
SESSION CHAIR: **James Villarreal, NASA/Johnson Space Center**

Advanced Network Architectures
(Paper not provided by publication date) 523

Sunspot Prediction Using Neural Networks 525

Behavioral Networks as a Model for Intelligent Agents 537

Autoclass II
(Paper not provided by publication date) 543

OMS FDIR-Initial Prototyping 545

SESSION 22: **Telerobotics and Supervised Autonomy**
SESSION CHAIR: **Dr. Antal Bejczy, NASA/Jet Propulsion Laboratory**

The Application of NASREM to Remote Robot Control 551

Technology for an Intelligent, Free-Flying Robot for Crew and
Equipment Retrieval in Space 559

The JPL Telerobot Operator Control Station-Operational Experiences 567

SESSION 23: **Intelligent Tutoring Systems**
SESSION CO-CHAIRS: **Dr. Michael Shafto, NASA/Ames Research Center**
Dr. Sherrie Gott, Brooks AFB

Training High-Performance Tasks with Intelligent Tutoring Systems
(Paper not provided by publication date) 577

An Intelligent Simulation Training System 579

Functional Description of a Command and Control Language Tutor 585

Success in Tutoring Electronic Troubleshooting 593

SESSION 24: Expert System Verification and Validation Panel
SESSION CHAIR: Peter A. Kiss/BDM International, Inc.
PANEL MEMBERS: C. Culbert, NASA/Johnson Space Center
K. Richardson, NASA/Ames Research Center
S. Johnson, NASA/Langley Research Center
R. Stachowitz, Lockheed
K. Bellman, Aerospace Corporation

No papers were presented at this session.

SESSION 25: KBS Standards
SESSION CHAIR: Dr. Chuck Hall, Lockheed AI Center, California

A Model for a Knowledge-Based System's Life Cycle 605

Knowledge-Based Systems (KBS) Development Standards: A Maintenance Perspective 611

Architecture Design and Implementation Standards for Medium-To-Large Scale Knowledge-Based Applications
(Paper not provided by publication date) 631

ACES - Space Shuttle Flight Software Analysis Expert System 633

SESSION 26: End Effectors/Manipulators II
SESSION CHAIRS: Captain Ron Julian, AAMRL/BBA, Wright-Patterson AFB

Test Bed Experiments for Various Telerobotic System Characteristics and Configurations 637

Needs and Uses of Human Engineering Data in the Design of a Force Reflecting Exoskeleton
(Paper not provided by publication date) 645

LTM-A-Dual-Arm Redundant Telerobotic System
(Paper not provided by publication date) 647

A Laboratory Breadboard System for Dual-Arm Teleoperation 649

SESSION 27: Human-System Interfaces
SESSION CO-CHAIRS: Dr. Betty Goldsberry, Lockheed Engineering and Sciences Company
Dr. Tim McKay, Lockheed Engineering and Sciences Company

Process and Representation in Graphical Displays 661

The Space Station User Interface Language: An Object-Oriented Language for Controlling Complex Systems
(Paper not provided by publication date) 667

Leonardo and the Design of Large Distributed Software Systems
(Paper not provided by publication date) 669

Evaluation of Off-Road Terrain with Static Stereo and Monoscopic Displays 671

SESSION 28: Fuzzy Logic
SESSION CHAIR: Bob Lea, NASA/Johnson Space Center

Ocean Feature Recognition Using Genetic Algorithms with Fuzzy Fitness Functions (GA/F^3) 679

Extensions of Algebraic Image Operators: An Approach to Model-Based Vision 687

Fuzzy-C Development System: A Complete Overview
(Paper not provided by publication date) 697

AUTHOR INDEX 699

KEYNOTE SPEECH TO SOAR '89 CONFERENCE

James B. Odom
Applied Research, Inc.

It is indeed a pleasure to be with you tonight because your field of endeavors of space operations, automation and robotics is a field of significant interest to me personally as well as one of the next new and real revenue source for our nation. I also find it very gratifying to see the excellent exchange between NASA and the Air Force at this and previous SOAR conferences.

I would first like to give you my perception of where our nation is or is not going in science and technology in the next few decades.

One should realize that within the span of just one human lifetime, we have moved from the sands of Kitty Hawk to the deserts of the moon; from the biplanes made of cloth and wood with no electronics to the world's first reusable spacecraft, the space shuttle that orbits the earth at 25 times the speed of sound with very advanced computers and electronic system.

While the nation, through NASA, has enjoyed phenomenal achievements and successes in its first 30 years of existence, it has also suffered a few significant setbacks, the most devastating being the Challenger accident. The agency has spent three painful but dedicated years recovering from and correcting the deficiencies that caused this failure. One should realize that any very complex system built by man that contains and expends the necessary energy required to quickly put the space shuttle orbiter into orbit will always be laden with some degree of risk. If we fly enough, there will likely be other problems. While the agency has done everything humanly possible to minimize the risk to the crews, the risk will never be zero. If it another problem should occur, we should not let it negate our resolve.

One should recognize that NASA set a launch schedule over two years ago to have launched two Technical Data Relay Satellites and one DoD mission and the Magellan spacecraft to the planet Venus by May, 1989. Those were all accomplished on schedule, demonstrating the space shuttle has resumed its planned flight program.

Under the capable leadership of Dr. Len Fisk, NASA will, in the next few months, embark on one of the most exciting and scientifically productive periods in the history of the U.S. space

program. During the next five years, we have scheduled 35 science and applications missions for launch of shuttle or expendable launch vehicles. This will represent the highest launch rate for science missions in the history of NASA. We are not simply launching small satellites as we did in the 1960's, but rather, we will be witnessing launches such as the Hubble Space Telescope, the Galileo, the Gamma Ray Observatory, the upper atmospheric research satellite. We will be producing science data at an unprecedented rate and quality, and our leadership in space science will not be in question. A number of the above spacecraft can only be built by our country at this time. This total complement of spacecraft will not only help us to better understand our universe, but more importantly, our earth and its related environment.

We are a nation that was founded by people who were determined to expand frontiers and to take advantage of that expansion to make the nation grow in wealth and strength. From the earliest days of the republic, our government has invested in this expansion. These actions are justified because of the potential to improve and/or enhance our standard of living!

I recently had occasion to read about the Lewis and Clark Expedition, the great exploration of the presidency of Thomas Jefferson, and I was struck with the parallels with our time.

When Jefferson was preparing his request for support for this expedition to Congress, he wanted to stress the geographic and scientific potential of the mission. However, he was advised that due to the frugal notion of Congress at the time - it seems that more than half of the federal budget was going to pay the interest on the national debt left over from the American Revolution - the journey could better be justified as an effort to explore the commercial potential of the territory, so when Jefferson defended his estimated $2500 cost for the expedition, it was "for the purpose of extending the commerce of the United States".

Now, this cost estimate of $2500. Well, it seems that first they underestimated by a factor of three the number of people required to execute the expedition. Then there were transportation delays; the boats were a new design and were several months late in delivery, and finally, the scientists kept insisting on adding new instruments for studying the environment. The final cost was not $2500, but close to $40,000. Not even the Galileo mission has had

that kind of overrun, but what a wise investment it was because it revealed the wonders of the west, and stimulated the eventual westward expansion of the population.

I could also not resist noting the magnitude of the investment our Government was prepared to make in the Lewis and Clark Expedition. A sum of $40,000, spread over the three years of the expedition, may not seem like much money, but compared with the federal budget of the time, it was considerable. There are many parallels between the Lewis and Clark Expedition and our space program today.

Following the Lewis and Clark expedition, our government invested in The Coastal Survey, many more exploration and Topographical Surveys, The Rail Roads, Panama Canal, Intercoastal Water Ways, Merchant Marines, Air Transportation Systems, Interstate Highway System and the Tennessee Valley Authority. One may argue the merits and pay back for any one of those, but they all have contributed significantly to the total fabric that weaved our nation's past. Our nation has not been one that was afraid to defend what we hold important or timid in our international roles. President Eisenhower notes in his first inaugural address in January, 1953, I quote "For history does not long entrust the care of freedom to the weak or the timid". Not only have we not been timid, we have been willing and desirable to be out front in science and technology. For many years, we were the "Big Boys" in many fields, but that role is fast changing. I grew up in the country near a great uncle who had the largest and best pack of fox hounds in that part of South Alabama. He had a saying I still remember. He would say "if you can't run with the big dogs, you should just lay on the porch". I fear we have too many of our population and leaders that are perfectly happy laying on the porch.

We, as Americans, still live in a country unsurpassed by none, but threatened by many. I really question our concerns as individuals. We have become so focused on the "short-term" in our personal and business investments, I believe we are losing our vision of the long-term concerns for our nation. We want to blame our leaders for our dilemma, but we must remember that <u>we</u> elected them and we <u>should</u> tell them what we think is really important to our nation. I think it is time we communicate our concerns to our leaders.

When one considers where our nation is or is not going, one is reminded of history's account of the directions the Ming Emperors took in the 15th century and the ensuing effect it had on world history. This account is summarized in an advertisement published by the Lockheed Corporation, which is as follows:

> *The Ming Dynasty and the Space Station. In 1492, China discovered Europe. This is the way history might have been written but for a sudden and fateful change of policy on the part of the Ming Emperors.*
>
> *In the early 15th century, China was the most advanced power on Earth. The Middle Kingdom led the world in such decisive technologies as gunpowder, printing, metallurgy, engineering, and medicine.*
>
> *Navigation was also in China's strong suit. From 1405 to 1433, vast Chinese fleets, which would dwarf much later European expeditions, were already exploring and opening up trade routes westward. By 1431, a fleet of sixty-two ships and 28,000 men had reached the east coast of Africa. It was only a matter of time before the Chinese discovered Europe.*
>
> *Then, in a sudden policy reversal, the Ming Emperors halted all further voyages and began to foster an atmosphere of xenophobic conservatism. Science and technology decayed. Trade became passive and the next five centuries saw China become one of the world's exploited nations rather than an exploring nation.*
>
> *The United States now also finds itself at a critical time in history. So many budget needs make investment in space exploration seem, by comparison, of lower priority. The funding of our orbiting Space Station, indeed, our whole space program, seems to be politically in doubt.*
>
> *One thing that is inevitable, though, is that humans will continue to reach out into space. It is in our nature and whether space is explored by the United*

Stated or the Soviet Union or Europe, it will be explored.

Had the Ming Emperors not hamstrung China five hundred years ago, history undoubtedly would have taken a dramatically different turn. But their moment was lost.

The direction the United States now takes into - or away from - space will have implications for our fate during the next five hundred years. Our moment is now.

President Elect John Kennedy delivered his last address to the Joint Houses of the General Court of Massachusetts on January 9, 1961. In it he said, "For what Pericles said to the Athenians has long been true of this commonwealth. We do not imitate - for we are a model to others". He continued to say, "history will not judge our endeavors and government cannot be selected merely on the basis of color or creed or even party affiliation. Neither will compliance and loyalty and stature, while essential to the utmost, suffice in times such as these".

"For of those whom much is given, much is required and when at some future date the high court of history sits in judgement on each one of us, recording whether during our brief span of service we fulfilled our responsibilities to the state, our success or failure in whatever office we may hold, will be measured by the answers to four questions:

First - Were we truly men of courage?
Second - Were we truly men of judgement?
Third - Were we truly men of integrity?
Finally - Were we truly men of dedication?"

These four characteristics are synonymous with the great men and women that have made our country what it is today. However, I fear these yardsticks are fast disappearing from our society that is producing our leaders for the future.

However, I would like to add that many of the individuals I know personally at this conference represent the best of the four criteria mentioned by John Kennedy. Not only does this group

represent some of the best forward thinking and planning people I know, but I also believe you are interested in what may very well be the next real new revenue stream for our nation. I believe you will also produce some of the most significant jumps in the productivity of our workforce, factories and space operation in areas such as:

1. Automation and Robotics (A&R) technologies have proven themselves in terrestrial/undersea applications within NASA (INCO), industry (Ocean Systems Engineering), and DoD; human-system interface a key element in successful applications.

2. NASA is properly concerned with the risks of introducing A&R technologies, but should be equally concerned with the risks/costs of not using them effectively; the "old way" of doing business has been shown to be very expensive and to inhibit productivity and technology upgrades. Unfortunately, one of your largest hurdles is getting by the old, relay, solenoid and stepper switch engineers and managers. You will literally have to skip one generation.

3. Knowledge of A&R capabilities and limitations essential to proper selection of applications and required tools; awareness and understanding of past successes/failures is an important starting point.

4. Implementation of A&R within major programs must have the commitment and support of senior and middle management to be broadly successful; bottoms up approach is easy to kill or of limited impact without support and motivation from above.

5. If an organization is not actively developing and transferring new technologies into its systems/operations, then it's actively going out of business. I look forward to the time when we will be starting new systems with expert and knowledge base systems an integral element.

6. NASA and the aerospace community must be aggressive in the broad application of A&R technologies to long-term programs such as SDIO and

Space Station, the impact of not doing so is severe (e.g., cost of operations, and inability to upgrade systems).

7. High productivity, low operations costs, and prevention of technological obsolescence essential for NASA programs, especially Space Station and any long lived program, which most DoD and NASA programs are today.

8. Mission requirements and resource limitations (especially crew time) produce a compelling case for the use of automation and robotics technologies in Space Station flight systems and ground operations.

9. The utilization and further development of large testbeds linked nationally will be a major contribution of this group gathered here this week.

I had to include Space Station because I truly believe the real progress in the applications of A&R will be directly proportional to the real productivity of the Space Station Freedom.

I am by the Space Station's Freedom Program as the 82 year old man by the name of Ish Iberstein that went to the Father and confessed of having an affair with a 25 year old young lady. The Father told him he would be glad to listen and help, but shouldn't be really be talking to his Rabbi. The old gentleman responded, "I will tell him, too, for I am telling everyone".

President Kennedy gave our country the last real challenge with his announcement for a national commitment to go to the moon and return safely within the decade. We, as a nation, celebrated the 20th anniversary of that great national success last week. We took an early lead in the so called "space race", but we have continuously relinquished that lead for the past 15 years.

Some of you may remember a quote Dr. von Braun made early in the manned space program. He said "We have not yet entered the Second Age of Discovery, the exploration of outer space. We are still in the harbor. We are still building and checking out the seaworthiness of our craft. We are still learning the things we need

to know about the new medium through which we shall have to travel"

"But we do stand on the threshold of the Second Age of Discovery. When the craft is ready and the oceans of space are calm, calm because we have learned the new medium and have prepared to sail on it, the new explorers will venture forth. The space-age Christopher Columbus and Megallan is presently unknown, but they are sitting somewhere today in a public school-house preparing for an adventure that exceeds their wildest day-dreams which today distracts them from their books".

Looking at NASA's future, it has been planning for many years, contrary to some public belief, for a reusable vehicle - the space shuttle (which now exists), a vigorous science and application program, (which now is existing or planned), and to have a permanently manned space station which will be both a research facility as well as the assembly hangar and launch pad for future lunar, planetary or deep space exploration. That program is now underway but is in serious question from a funding standpoint. We have convinced Canada, Europe and Japan to invest $8 billion, coupled with our $16 billion, to make this next logical step an international reality of the major free world nations. I fear this program will be lost or scaled down to the point where our international partners will decide we are an unreliable partner and will continue on their own or with Russia. This reduction, I believe, would be a dreadful mistake on our nation's part. I hope we are not repeating the Ming Dynasty mistake.

One of the best kept secrets in our country is the excellent research conducted by our small business. I have had numerous opportunities in the past few months to see some of it first hand. The company, Applied Research, that I am now associated with, has developed many excellent optical weld control systems that close the loop between a vision system and the weld torch, voice control systems, low cost parallel processors, and video image systems just to mention a few. Fortunately, that is occurring throughout our nation.

We frequently hear questions as to why we have to do our space exploration with humans versus unmanned spacecraft. Much can and will continue to be accomplished unmanned. I like a recent Congressman's answer to this question. His response was that he

could not recall seeing a single picture or account of a Conostoga Wagon going west that was unmanned. I believe that when we, as humans, cease to wonder what is over the next hill or what makes up our earth's planetary system or what our universe consists of, we will probably not long exist as a nation.

While our nation has enjoyed a technological position of leadership since World War II, that position is no longer guaranteed. One only has to look at a few recent and near planned events to have cause to worry.

The planned industrial and financial unification of Europe set for 1992 will be unprecedented in history and will place them in a much stronger industrial posture to compete with the U.S. and Japan. Japan has taken our research and beaten us to the market at a rapid rate. They have increased their expenditures going into basic research, which they have not done in the past, and their institutions of higher learning have increased from 50 at the end of World War II to over 550 today. Given these facts, their plan and stated intention to continue to beat us to the marketplace with new and better products should be cause for grave concern to us. While Japan and Europe are investing more in R&D, we as a nation are investing less than one fourth the percentage of our nation's total budget today compared to the time our nation was going to the moon 20 years ago. Three short years ago, we were commanding 87% of the world market in commercial aircraft, which resulted in a positive balance of trade in the amount of 17 billion dollars. This year, that percentage of the world market may drop to approximately 50% and an attendant drop will occur in this our nation's best contributor to the positive trade balance. This, too, should give us cause for concern, but does it?

Another area for concern is the quantity of engineers and scientists that will be graduating from our U.S. schools and institutions by the year 2000, just 11 short years from now. It is projected that the U.S. will have a shortage of engineers and scientists in the amount of 500,000 by the start of the next century. Further, the percentage of foreign nationals that will be trained/educated by our institutions (especially advanced degrees) versus U.S. citizens also gives cause for concern.

I truly fear we have created a congress that is disproportionately involved in issues like, "Is it ok to pray in school

or to burn our beloved flag?" The latter one does not take me long to decide and I do not need a constitutional amendment to settle that one. We also see our congress getting bogged down in ethical investigations while at the same time trying to legislate morality in the nation's workforce. I would suggest that morality starts in the home, the family unit that has taken a severe beating in the 60's and 70's. We have recently gone from a nation for freedom of religion to one of freedom from religion.

God has abundantly blessed our nation and made us stewards of the greatest set of resources any nation possesses. If we are to continue as a leader nation, we must not only put to good use the resources we have, but look for new revenue sources for the future. I believe in the necessity for continued investment in science and technology if we are to find cures for human diseases, protect and clean our environment and maintain our posture in a very technologically oriented society.

We, as a nation, must seriously ask ourselves where we will make our investments in the future. I believe the government should continue to take the lead in large long-term R&D programs with industry taking the resultant products to the marketplace. The advanced technology posture of the U.S. aerospace program can be improved significantly with the fostering of an attitude of cooperation and trust between U.S. government, industry, and university participants. This teamwork/cooperation could be attained and/or improved with minimum impact to the national budget.

History has afforded the leadership of the 18th century to the French, the 19th century to the British, the 20th century to the Americans. I, for one, would like to make it two in a row.

I commend you and thank you here tonight for your vision, your dedication and willingness to step out into the cutting edge of technology for <u>this country</u>.

ARTIFICIAL INTELLIGENCE

WITHIN

AFSC

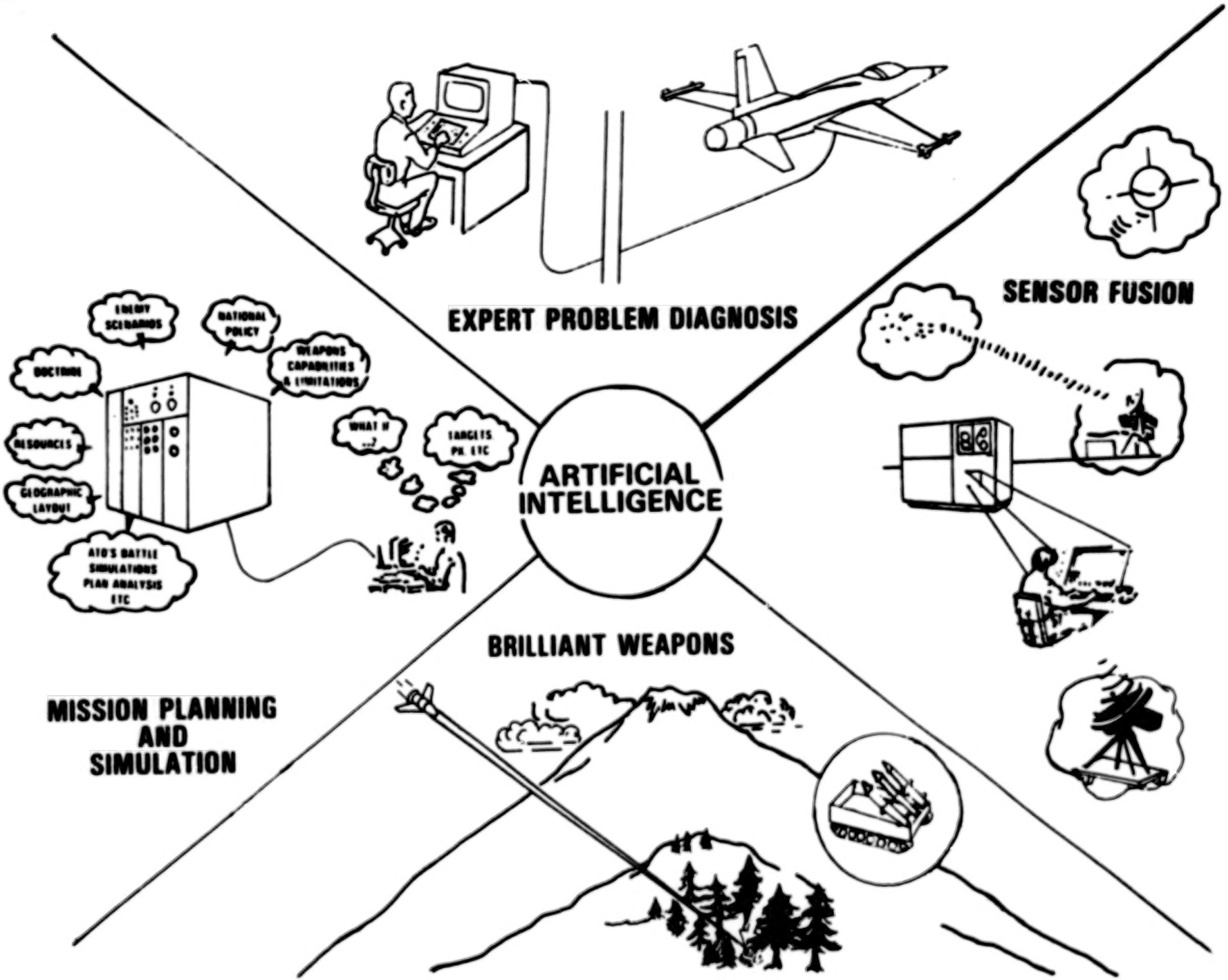


HEADQUARTERS AIR FORCE SYSTEMS COMMAND
ANDREWS AFB DC 20334-5000

MARK A. GERSH, CAPT., USAF
CHIEF, ARTIFICIAL INTELLIGENCE TECHNOLOGY DEVELOPMENT
DCS/TECHNOLOGY & REQUIREMENTS PLANNING

HQ AFSC/XTKC
ANDREWS AFB DC 20334-5000

OFFICE PH (301) 981-2883
AUTOVON 858-2883


EXPERT PROBLEM DIAGNOSIS
SENSOR FUSION
ARTIFICIAL INTELLIGENCE
BRILLIANT WEAPONS
MISSION PLANNING AND SIMULATION
ENEMY SCENARIOS
NATIONAL POLICY
WEAPONS CAPABILITIES & LIMITATIONS
DOCTRINE
RESOURCES
GEOGRAPHIC LAYOUT
ATO'S BATTLE SIMULATIONS PLAN ANALYSIS ETC
WHAT IF ...?
TARGETS PK ETC

AI

&

THE AIR FORCE

OFFICE OF SCIENTIFIC

RESEARCH



15

AIR FORCE AI BASIC RESEARCH

- LARGE KNOWLEDGE BASE/INTELLIGENT RETRIEVAL
  - ANALOGICAL/CASE-BASED REASONING
  - UNCERTAINTY
  - QUALITATIVE REASONING
  - INFORMATION INDEXING
- MACHINE VISION/IMAGE UNDERSTANDING
  - 2-D IMAGES
  - FEATURE COMBINING
  - LOW-, MID-, HIGH-LEVEL VISION ALGORITHMIC MODELING

RESEARCH THRUSTS (CONT.)

- PROBLEM SOLVING UNDER RESOURCE CONSTRAINTS
  - ACCURACY
  - TIMELINESS
  - COMPLETENESS
- MIXED SYMBOLIC AND NUMERIC COMPUTING
  - REASONING ABOUT PHYSICAL SYSTEMS (SYSTEMS WHICH REQUIRE QUALITATIVE AND QUANTITATIVE ANALYSIS)

MINOR AI THRUSTS

1. MACHINE LEARNING
2. PLANNING AND PROBLEM SOLVING
3. MASSIVELY PARALLEL SYSTEMS
4. APPROXIMATE REASONING
5. NATURAL LANGUAGE

AI

&

THE ROME AIR

DEVELOPMENT CENTER

Communications Directorate

Communications Network Management

Cooperative distributed problem solving applied to the management and control of communication networks.

Distributed Intelligent Systems for Communication Network Management - Expert system technology applied to traffic monitoring and control for communications networks.

Automated Services for Inter-Personal Communications

Expert systems for communication among dispersed C2 centers.

Network Security Guards

Automated network interfaces for systems of varying security classification levels.

Sentient Radio

Intelligent adjustments for improved radio reception.

Reliability and Compatibility Directorate

Smart Built-In Test and Time Stress Measurement Devices

Develop and implement integrated failure detection and diagnosis mechanisms for electronic systems. Collection, analysis and storage of real-time environmental stress data.

Circuit Diagnostics and Testing

Diagnosis of failures in electronic systems across all levels of maintenance.

Computer-Aided Design for Testability and Reliability

Computer-aided tools for design and testing of electronic circuits and components.

Intelligent Design for Testability - Develop tools for use in conjunction with CAD/CAE package to design electronic circuits or systems with a high inherent level of testability.

Intelligence and Reconnaissance Directorate

Speech Processing and Understanding
Provide information processing assistance for intelligence analysis.

Knowledge-Based Image Interpretation
Advance the state-of-the-art in digital image exploitation.

Automated Database Update
Machine understanding of English text reports of events for database and knowledge base update.

Language Interface Module (LIM) - Integration of natural, human speech and free-form text inputs.

Neural Networks Applied to Model-Based Systems
Improving performance of model-based systems through training from Neural Networks.

Command and Control Directorate

Planning

Support the rapid, accurate, and efficient creation and modification of plans.

Time-Oriented Problem Solving - Develop problem solving theories about interacting with external events and other intelligent agents.

Knowledge-Based Problem Solving

Improve parallel, distributed, and real-time problem solving technology.

Knowledge-Based Simulation

Develop simulation technology that is easier and less costly to use, and more representative of real-world situations.

Knowledge-Based Expert Systems

Develop component technologies supporting the next generation of knowledge-based expert systems, and applications employing those capabilities across the spectrum of C3I domains.

Knowledge Base/Database Architectures

Integration of database and knowledge base technology to provide real-time processing of very large knowledge bases.

Command and Control Directorate

Intelligent Interfaces

Make increasingly complex C3I systems more intuitive, and provide intelligent assistance in their use.

Intelligent Man-Machine Interface Evaluation - Facilitate the accessibility and application of current and developing interface technology.

Parallel AI

Identify and exploit parallelism in knowledge-based problem solving.

Software Engineering

Improve software analysis, specification, validation, quality and productivity.

Knowledge Based Software Assistant (KBSA)

Develop a knowledge-based lifecycle paradigm for the design, development, and support of very large software projects.

Surveillance Directorate

Optimal Filtering for Signal Processing

Extracting information from incoming signals for situational and behavioral assessment, system response, and optimal resource management.

High-Level Adaptive Signal Processing - Signal processing testbed for electromagnetic type localization and classification using time-frequency domain feature extraction.

Low Observable and Multitarget Tracking

Target tracking with multispectral knowledge-based surveillance systems.

Multispectral Data Fusion

Associate data from different sensors to increase coverage of low observables.

Knowledge-Based Signal Processing and Data Fusion - Define optimizations for merging numeric and symbolic signals from multiple distributed systems.

Electromagnetic Sciences

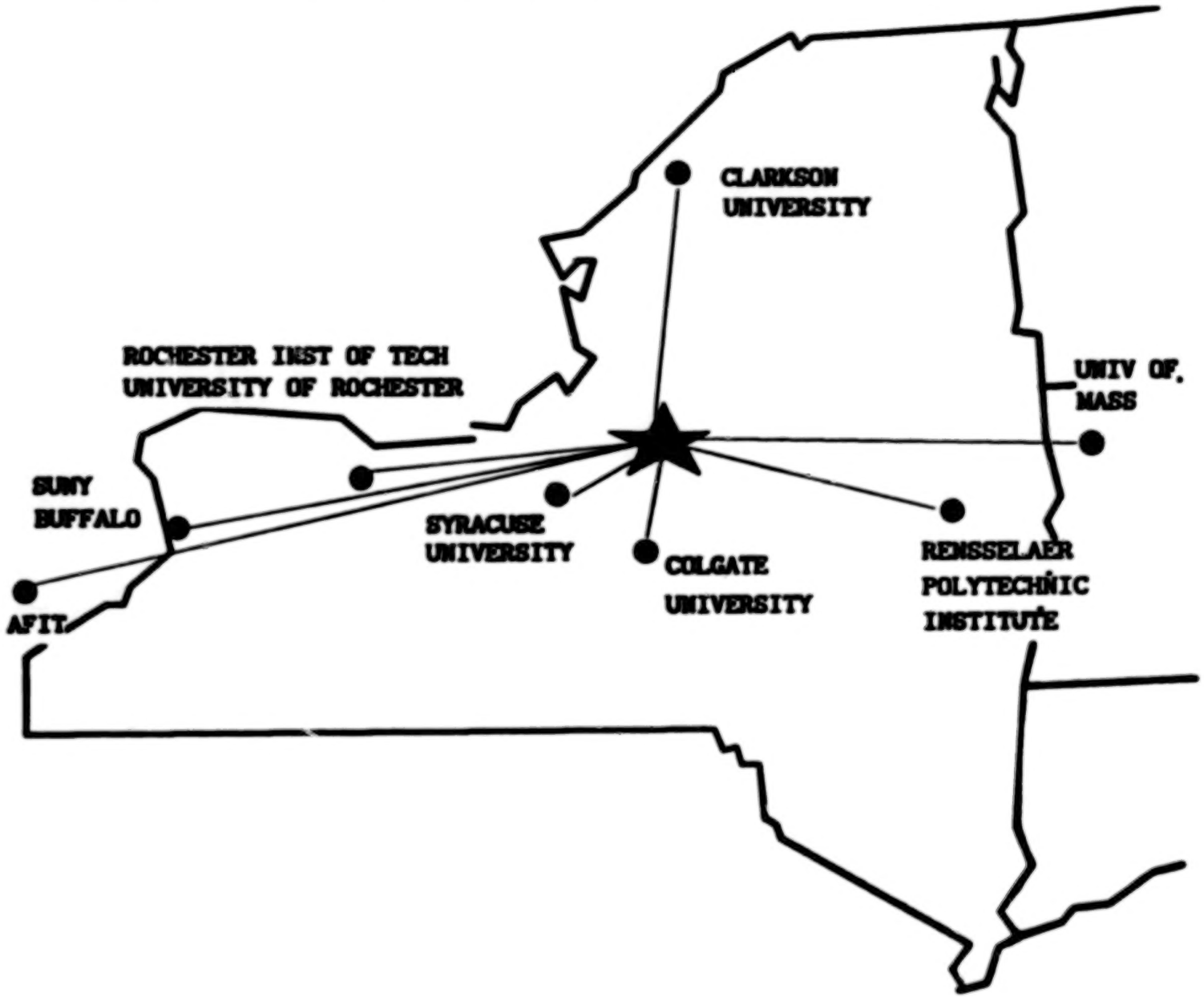
Intelligent Antennas

Processing signals for phased-array antennas.

Neural Networks Applied to Signal Analysis

Regeneration of text from corrupted channels.

RADC ARTIFICIAL INTELLIGENCE UNIVERSITY CONSORTIUM



NORTHEAST AI CONSORTIUM

UNIVERSITY	TECHNICAL PROGRAM AREA
CLARKSON	COMMUNICATIONS SYSTEMS CONTROL
RIT	SPEECH UNDERSTANDING
UNIVERSITY OF ROCHESTER	TIME-ORIENTED PROBLEM SOLVING
RPI	PHOTO INTERPRETATION
SUNY BUFFALO	EXPERT SYSTEMS FOR MAINTENANCE
SYRACUSE	INTELLIGENT KBSs, ADVANCED LOGIC PROGRAMMING ENVIRONMENTS
UNIVERSITY OF MASS	AUTOMATED MULTI-SOURCE KA

AI

&

THE SPACE

TECHNOLOGY

CENTER



REQUIREMENTS

INTELLIGENT CONSOLE FOR SATELLITE CONTROLLERS

TRAINING AIDS FOR CONTROLLERS

FOG PREDICTION AT LAUNCH SITES

SATELLITE AUTONOMY

FAULT DIAGNOSIS

RANGE SCHEDULING

V & V OF EXPERT SYSTEMS

ENVIRONMENTAL PROBLEM IDENTIFICATION

AUTONOMY FOR NAVIGATION AND CONTROL

REDUCTION OF MANPOWER REQUIREMENTS

AFSTC AI INITIATIVES

HQ SPACE TECHNOLOGY CENTER

1. EXACT (SBIR ON SATELLITE & NETWORK CONTROL)
2. EXPERT SYSTEMS FOR 1750A ARCHITECTURE
3. V & V FOR EXPERT SYSTEMS
4. AI TECHNIQUES FOR SATELLITE AUTONOMY
5. AUTONOMY TECHNOLOGY SEGMENT

THE WEAPONS LAB

6. RELATIONAL DATABASE NATURAL LANGUAGE INTERFACE
7. NEURAL NET PATTERN RECOGNITION

THE GEOPHYSICS LAB

1. FOG PREDICTION
2. TURBULANCE ADVISOR
3. NEURAL NETWORK CLOUD CLASSIFIER (SATELLITE IMAGERY)

THE ASTRONAUTICS LAB

4. EXPERT SYSTEM FOR NON-DESTRUCTIVE EVALUATION
5. EXPERT SYSTEM FOR ENGINE DESIGN

AI

&

THE HUMAN RESOURCES

LABORATORY



HUMAN RESOURCES LAB AI THRUSTS

Conducting research to investigate issues involved in applying AI technology to training systems

- TOOLS

Centers on the construction of tools which facilitate the design, development, and coding of Intelligent Tutoring Systems (ITS)

- TUTORS

Development of ITSs within the context of Air Force technical training

- TESTBEDS

Systematically vary ITS designs to investigate the instructional effectiveness of the different design approaches

AI

&

THE ARMSTRONG AEROSPACE

MEDICAL RESEARCH LAB



MAJOR EFFORTS

CURRENT PROJECTS

- Manned threat evaluation and modeling
- Flight performance evaluation
- Hardware Neural Nets
- Noise cancellation
- Auditory signal processing

PROPOSED

- Knowledge Workshop

AI

&

THE ARMAMENT LAB



AFATL AI INTEREST

MUNITIONS

ADVANCED FUZING
EXPLOSIVES DESIGN SUPPORT
WARHEAD DESIGN SUPPORT

ADVANCED GUIDANCE

IMAGE PROCESSING
AUTOMATIC TARGET RECOGNITION
REAL TIME INFERENCING
MUNITION/AIRCRAFT COMM

AEROMECHANICS

DATA ANALYSES FOR GUN TUNNEL
A/C INTERFACE CERTIFICATION
XRAY IMAGE ANALYSES
COMPUTATIONAL FLUID DYNAMICS
(GRID GENERATION, MODELING)
MUNITIONS MAINTENANCE AIDS

ANALYSIS AND STRATEGIC DEFENSE

IMAGE PROCESSING
REAL TIME INFERENCING
AUTOMATIC TARGET RECOGNITION

AI

&

THE WRIGHT RESEARCH

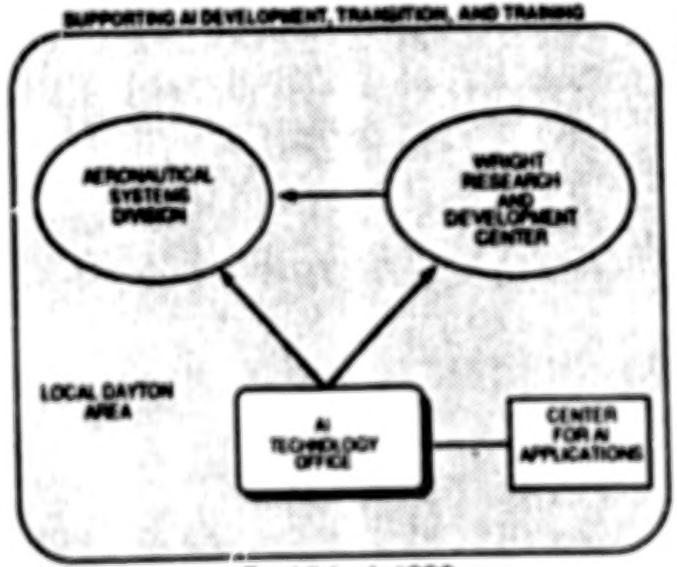
and

DEVELOPMENT CENTER



MISSION

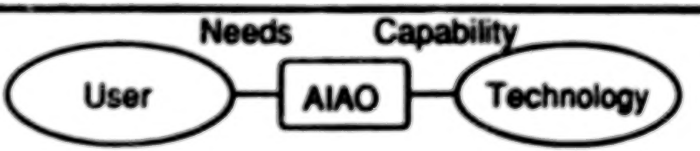
- Focus and facilitate AI related efforts within ASD and Wright Research and Development Center
- Address high priority aerospace weapon system requirements
- Provide emphasis on near-term AI technology transition opportunities
- Provide AI education and training



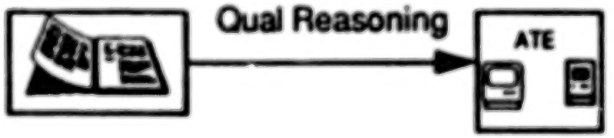
Established: 1986

APPROACH

- Technology Assessment



- Technology Transition



- In-house Capability Enhancement



- Advocacy

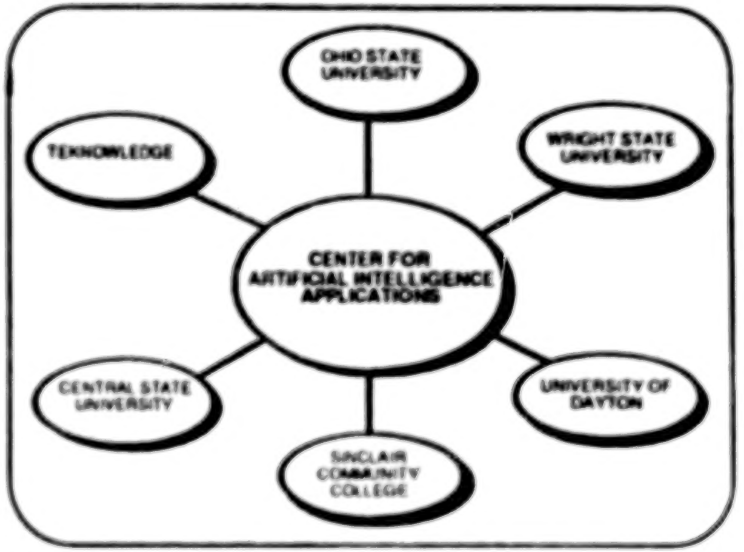


- Dayton AI Center



DAYTON CENTER FOR ARTIFICIAL INTELLIGENCE APPLICATIONS

- **Miami Valley Research Institute**
  - Five local universities
  - Teknowledge
  - Local companies



- **Objective**:
  - Expand the local AI talent base
  - Study the application of AI technologies to USAF
  - Foster AI technology transition
  - Provide AI education and training

WRDC AI THRUSTS

COCKPIT INTEGRATION

1. LEARNING SYSTEM, PILOT AIDING
2. PILOT'S ASSOCIATE DISPLAYS/CONTROLS
3. INTELLIGENT PILOT-VEHICLE INTERFACE
4. THREAT EXPERT ANALYSIS SYSTEM
5. PILOT'S ASSOCIATE PROGRAM
6. CREW-AIDING PROCESSOR AND DISPLAY SYSTEM
7. V & V OF AI SOFTWARE

43

AERO PROPULSION AND POWER

1. NEURAL NETWORKS FOR TURBINE ENGINE DIAGNOSTICS
2. PARAMETRIC TRAJECTORY ANALYSIS

MATERIALS

3. QUALITATIVE PROCESS AUTOMATION
4. FEATURE BASED DESIGN

FLIGHT DYNAMICS

5. DIAGNOSTIC ABDUCTIVE AND INDUCTIVE REASONING
6. SPACE TRANSPORTATION
7. FLIGHT CONTROLS (GROUND-BASED, IN-FLIGHT, SELF-REPAIRING DIAGNOSTICS)

AVIONICS

- MACHINE INTELLIGENCE
  - NEURAL ADAPTIVE LEARNING
  - MACHINE PERCEPTION & IMAGE UNDERSTANDING
  - INTELLIGENT AVIONICS
- AI PROCESSOR DEVELOPMENT
- MODEL BASED VISION

SOAPBOX ISSUES

- COMMUNICATION & COOPERATION
- REQUIREMENTS
  - PURE RESEARCH: WHAT DO WE NEED FROM AI BEFORE IT CAN BE APPLIED?
  - APPLIED RESEARCH: WHAT DO I NEED FROM AI BEFORE IT CAN BE APPLIED IN MY DOMAIN?
- PROVE YOUR PUDDING

PRELIMINARY TESTS OF AUTOMATED ASSEMBLY OF SPACE STRUCTURES

R. Will
NASA/Langley Research Center

(Paper not provided by publication date.)

EXPERIMENTAL RESULTS OF INTEGRATED EVA/TELEROBOTIC WORK SITES

D. Akin, J. Hedgecock, and E. Sorenson

(Paper not provided by publication date.)

Mobile Transporter Path Planning

Paul Baffes and Lui Wang

Artificial Intelligence Section, FM72
Johnson Space Center, Houston, Texas 77058

1. ABSTRACT

(The work presented here is a copy of a presentation given at the SPIE cambridge symposium in November 1988). The use of a genetic algorithm for solving the mobile transporter path planning problem is investigated. The mobile transporter is a traveling robotic vehicle proposed for the space station which must be able to reach any point of the structure autonomously. Elements of the genetic algorithm are explored in both a theoretical and experimental sense. Specifically, double crossover, greedy crossover, and tournament selection techniques are examined. Additionally, the use of local optimization techniques working in concert with the GA are also explored. Recent developments in genetic algorithm theory are shown to be particularly effective in a path planning problem domain, though problem areas can be cited which require more research.

2. INTRODUCTION

The current design of the proposed space station involves a rather large, skeletal truss structure to which various living modules, laboratories, and equipment will be attached (see Figure 1). To facilitate the performance of maintenance tasks and the transportation of materials around the station, a mobile transporter (MT) system is being designed which will be capable of traversing the truss structure. At the very least, as the MT moves it must be able to avoid collisions with the objects attached to the truss, and it is obviously desirable that a path of shortest distance be selected before moving the MT between any two points. Since the configuration of the space station is modifiable and the MT must be able to begin its trek from any point on the truss structure, the selection of a good path for the general case is a difficult task.

Planning a path for the mobile transporter falls into a category of optimization problems known as trajectory planning problems. Typically the solution spaces for these problems are multimodal, i.e., they are characterized by many local maxima at which acceptable solutions may be found. Classical gradient techniques for solving such optimization problems have not proven effective due to difficulties in selecting relevant features from the domain from which to build a hill climbing algorithm. Several studies such as one conducted by Gomez-Tierno[1] have suggested that genetic algorithms (GA) offer more promise due to the algorithm's global approach to problem domains. These studies also suggest that more theoretical work on GAs needs to be done before the technique can be satisfactorily applied to trajectory planning. However, recent advances in GA theory have directly addressed problems, such as path planning, whose solutions are order-dependent. What follows is an examination of the application of current GA theory to the mobile transporter problem. The following specific issues are addressed: (1) the details of casting the mobile transporter problem into a genetic algorithm format, (2) predictions of pitfalls or enhancements which can be derived from GA theory, and (3) discussions of some issues still open for using a GA to perform path planning for the mobile transporter.

3. BACKGROUND

3.1 GENETIC ALGORITHMS

It is assumed that the reader is familiar with the various elements of the genetic algorithm and an introduction is not provided here. However, a short review of some of the theory and terminology of GAs is presented as background material for the sections which follow. Recall that the recombination process of a GA typically involves two operators: *crossover* and *mutation*. A crossover occurs between two solutions, called *parents*, which are probabilistically selected from the

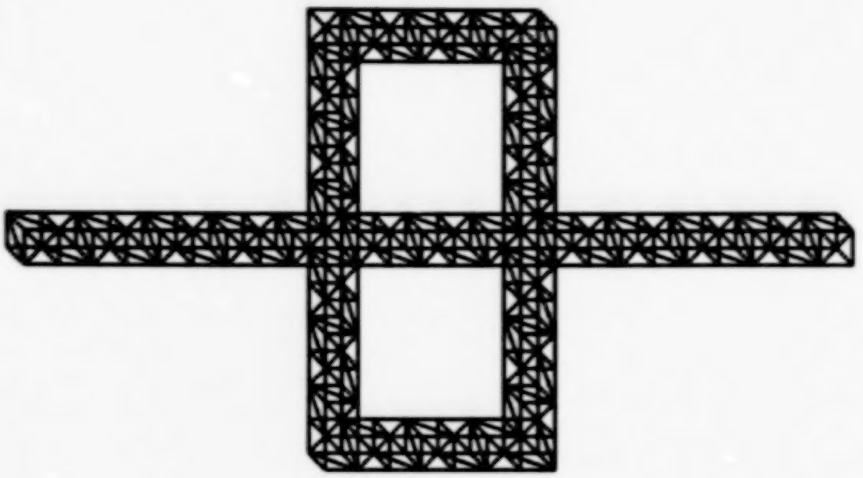


Figure 1. Empty space station truss structure

population based upon their fitness values. A *crossover point* is chosen at random between zero and the length of a solution string, and the two parents are split at that point into front and back ends. The back ends of the two parents are then interchanged and reconnected, creating two new solutions which have roughly half of their values from each parent. Mutation, on the other hand, occurs within a single parent and operates locus by locus. Should a mutation occur at a particular locus point, a new allele value for that point is randomly chosen. The result is a new solution which differs only slightly from the original. While other genetic operators are sometimes used, these two represent the minimum necessary for the correct operation of a GA and are the most widely used.

Theoretical formulations of genetic algorithms have been developed around the concept of a *schema* which represents a partition of the solution space. Roughly speaking, a schema can be thought of as a substring of a solution with some elements given specific values and some left undetermined. For example, given a binary set of allele values (i.e. 0 and 1) a typical schema might look like the following:

*10***1**

where the "*" symbol indicates that either allele value may occur at the given locus. Consequently, the schema may be thought of as representing groups or hyperplanes of substrings which specify particular patterns of allele-locus combinations. Schemata have two important characteristics, their *order* and their *defining length*. The order of a schema is the number of its specified elements; the above schema has an order of 3. The defining length is defined as the distance *between* the outermost specified elements of a schema. In the above example, if we were to number the loci shown we would have 9 loci with the two "1" values occurring at positions 2 and 7. Taking the difference of these yields a defining length of 7 - 2 or 5. Note that this is one less than what might normally be considered the size of the schema, sometimes called the *length* of the schema, which here would be 6. In this paper the defining length will be specified by ∂ (delta) and the length will be specified by D such that $D = \partial + 1$.

Holland's[2] work showed that, given the proper problem domain, genetic algorithms will evolve toward nearly optimal solutions by building up complicated structures out of small components which can be described as schemata. A typical expression of this phenomenon as described by Goldberg[3] would look like the following:

$$n(H,t+1) \geq n(H,t)\,\frac{f(H)}{\bar{f}}\left[1 - p_c\,\frac{\partial(H)}{L-1} - p_m\,o(H)\right] \qquad (1)$$

where H stands for a particular schema, $n(H,t)$ is a measure of the number of copies

of the schema in the population at time t, f is the fitness function, L is the length of a solution, p_c and p_m are the probabilities of crossover and mutation, and ∂(H) and o(H) are the defining length and order of the schema. The fractional term indicates that the number of schema in the population will change proportional to the fitness of solutions containing the schema relative to the average fitness of the population. Of particular interest is the last term of the equation which is a measure of the extent to which a crossover or mutation is likely to disrupt the schema. Note that the greater the defining length or order of the schema, the greater the chances that it will be altered by a crossover or mutation, thus driving down the number of schema in the population. Consequently, schemata which are *short* and of *low order* are probabilistically favored by the genetic algorithm, and any GA which is to work effectively must contain operators which prefer such small, low order building blocks.

Recent theoretical work by Goldberg and Lingle[4] is centered around the idea of a double crossover technique termed a partially matched crossover (PMX). The PMX operator is particularly effective for domains in which ordering of values is essential. In such problems, both the allele values and their positioning relative to other alleles contribute to the overall fitness of a solution. An example of an ordering problem cited by Goldberg and Lingle is the traveling salesman problem (TSP) in which a salesman must visit some number of cities in a circuit while taking the minimum distance to do so. Crossing over two solutions using the PMX operator begins by selecting two crossover points, instead of just one, which define a *cut* of length K covering all alleles occurring between the two crossover points. The cut regions of the two solutions to be crossed are then exchanged. Any duplications which may result *outside of the cut regions* are swapped between the two new solutions to enforce the constraint of exactly one occurrence of each city. Note that no swapping is allowed to occur inside of either cut, thus preserving whatever order existed among the alleles within that region. Goldberg and Lingle present the survival chances of a schema using the PMX as

$$P(S) = \frac{K-D(H)+1}{L} + \frac{L-K-D(H)+1}{L}\left[1 - \frac{K+1}{L}\right]^{o(H)} \qquad (2)$$

where K is the length of the cut, D(H) is the length (not defining length) of the schema, o(H) is the order of the schema, and L is the length of a solution. The two terms of this equation represent the two conditions under which a schema may survive a crossover: entirely within the cut region or entirely outside of the cut region. Schemata which span either of the two crossover points are assumed to be disrupted. The first term of the equation, for schema inside the cut, is only dependent upon the length of the schema. Again, short schema are favored. The second term, for schema outside of the cut, is dependent upon both the length and the order of the schema. The second factor of this term is a measure of the chances that any single locus outside of the cut will survive the swapping phase of the PMX. Since this chance is the same for all specified bits of the schema, the more specified bits the schema has the lower its chances of survival. Thus the lower order schema are also favored by the PMX operator.

To summarize, this review has presented two formulas which show the probability of schema survival. The first covers single crossover cases and the second can be used for the PMX crossover operator. The analysis section which follows (section 4) will discuss how these formulas relate to the application of GAs to the mobile transporter problem.

3.2 THE MOBILE TRANSPORTER

Several features unique to the mobile transporter's design and operating environment are salient to the use of genetic algorithms as a modeling approach. As it is currently envisioned, the MT will be able to transport heavy loads (up to 300,000 lbs.), which will substantially hinder its rate of travel. It is estimated that the travel time between the two most remote points of the station may be as much as 6 hours. Furthermore, since the truss structure of the space station is 3-dimensional, the MT will have at least two types of movement:

along the surface of one plane, or between two planes. The latter is a much more complex maneuver lasting approximately five times the duration needed to travel an equivalent distance along a single plane. Due to these constraints, path planning for the MT must be efficient and adaptable so that changes in the scheduled uses of the MT may be quickly addressed by an appropriate change of path.

However, trajectory planning for the MT does not map directly into a genetic algorithm format. One of the first obstacles encountered is the need for a *variability of solution length* which does not occur in a classical GA design. The paths developed for the MT vary in length not only between problems as the MT is assigned different starting and ending points, but also within a single problem as the GA is running. Furthermore, the existence of variable length solutions complicates the structure of the crossover operator, since some strings will have more crossover opportunities than others and since randomly chosen crossover points will almost certainly produce invalid solution paths. To ensure that a crossover produces a valid result, one can use an *intersection crossover*. An intersection crossover is identical to the traditional crossover except for a preceding step to compute the set of intersecting points for the two parents to be crossed. Once calculated, this set is used as the basis for the selection of a crossover point, thus ensuring that the substrings of the two parents used to make the new solution will be rejoined at a common point. Note that the use of the intersection crossover has two potentially adverse effects: the selection of a crossover point is no longer a uniformly random choice (those points that are more likely to be present in a path will be chosen more often) and the probability of a crossover occurrence is diminished because some parent pairs may have an empty intersection set. Finally, observe that double crossovers performed using the set of intersecting points can produce cycles in the resulting solutions, even if the parent paths of the initial population of solutions are constrained to be free of cycles. Such loops in the resulting solutions must be removed if the constraint of no-cycle paths is to be maintained in subsequent generations.

A *delooping* procedure which will accomplish this can be defined as follows. Starting with the beginning of the path, a candidate point is chosen and all points following the candidate are visited in turn. If any of these following points matches the candidate, all elements of the path between the candidate and the matching point are removed from the path. The checking then continues with the same candidate until the end of the path is reached, at which time the point currently following the candidate is chosen as the new candidate. The process is repeated for all points until the end of the path is reached. Note that no more than two matches will be found for any given delooping operation. This is due to the fact that both the original path and the region spliced in from the other parent are guaranteed to have no cycles. As a consequence, no two matches in the crossed over solution will be identical and all such matches will occur between a point inside of the cut region and a point outside of the cut region. Since delooping removes everything between two matched points, a single delooping action must cross one of the two crossover points thereby removing it from the solution. With only two crossover points in a double crossover, only two such delooping actions can occur.

Thus, several features of the mobile transporter path planning problem make it incompatible with classical GA techniques. In particular, the existence of varying length paths gives rise to a need for a different mechanism for the crossover operator. An intersection crossover is defined to ensure that crossing over will produce valid paths, but this in turn gives rise to a cycling problem when two crossover points are used. Consequently, a delooping procedure has also been developed to rid a path of cycles. The resulting crossover operation becomes a combination of intersection crossover followed by the delooping procedure. However, a question remains open as to the possible adverse effects of these new operators on the functioning of the resulting GA.

4. APPROACH

A twofold approach was taken to investigate the translation of the mobile

transporter problem into the format of a genetic algorithm. First, an examination of the current GA theory was made in an effort to predict the behavior of the resultant system. Second, several simulations were written to collect experimental data on the various GA methods under question. The focus of these two approaches was aimed at the following topics:

1. *Single vs double crossover.* It was felt that the MT path planning problem was significantly order-dependent, but the degree to which this was true was unknown. It was questionable whether a single crossover would be sufficient for a problem which exhibited only a small degree of dependence upon the order of its values and, conversely, whether or not a double crossover in such a situation would be disruptive.

2. *Effects of intersection crossover and delooping.* Given that a new crossover operator was needed for performing double crossovers, it was not immediately clear what effect this would have on the GA's performance. The question arose as to whether the new crossover operator might adversely effect the diversity of the population due to some points in the domain being more likely candidates as crossover points.

3. *Use of a greedy crossover.* Liebens et al[5] describe a process whereby a greedy algorithm technique was employed during the selection of crossover points which greatly enhanced the conversion of the GA to an optimal answer. The extent to which this idea would help in the MT problem was an intriguing question.

4. *Need for a mutation operator.* Because the GA described here used a delooping procedure rather than the swapping procedure of the PMX, it was unclear to what extent this might effect the preference for low order schema. Recall the presence of an order-dependent factor in the second term of equation (2). Since swapping was replaced by delooping, this factor would not present in the GA for the mobile transporter, resulting in the loss of the only order-dependent term of the equation. If a mutation operator could be devised which would return such a factor to the equation, the question then became whether to allow mutations to occur within the cut region of the double crossover or to force all such.changes to occur outside the region as is done with the PMX operator.

5. *Local optimization.* During our simulations, we found that the GA seemed to converge before the last few "kinks" of a path could be straightened out. Consequently, our analysis included an examination of the extent to which local optimization techniques could be used to enhance the overall performance of the GA.

6. *Tournament selection.* Normally, parents are chosen for recom-bination based on a roulette-like approach where each parent occupies a percentage of an imaginary roulette wheel proportional to its fitness. Parents with the highest fitness values are chosen most often. However, this method of selection has two disadvantages. If fitness values vary widely, a super-fit parent can easily dominate the population and drive the GA to premature convergence. ON the other hand, if fitness values are clustered, small differences which may be important are not emphasized enough. Tournament selection provides an alternative to roulette selection which avoids both of these problems.

5. EXPERIMENTAL RESULTS

Much of what was expected was borne out by the simulation results. Three different simulations were developed to test the issues cited in section 4 above: a simulation of the mobile transporter, a reproduction of the experiment performed by Goldberg and Lingle for the TSP problem, and a simple scheduling simulation. The results for crossover operators are presented first, then the tests involving the mutation operator, and finally the results of using local optimization and tournament selection.

5.1 Crossover

Figure 2 shows a comparison between typical single and double crossover runs for the mobile transporter genetic algorithm. The crossover technique used for this simulation was the intersection crossover followed by the delooping procedure described in section 3.2. Note that the descending fitness values of subsequent generations indicate that the path length is improving by getting shorter. As expected, the MT problem exhibits enough order-dependence to benefit from the use of a double crossover.

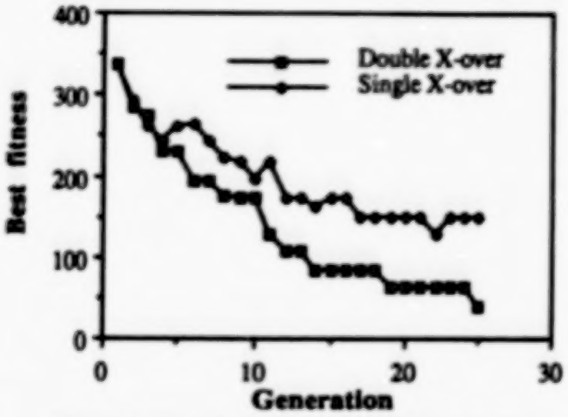


Figure 2. Best fitness for single/double crossovers

Moreover, the double crossover proved to have no adverse effect in the simple scheduling problem. In this simulation, four fictitious employees were to be scheduled over a three week period with one employee working per day. Each employee was required to work between some minimum and maximum number of days, with some employees working less than others. Also, no employee was allowed to work two days in a row. The only ordering constraint introduced into the problem was a requirement that one particular employee always work before another employee each week, though no specification was made as to how far apart the two employees had to work. Fitness values were determined by counting the number of constraints violated by a solution, thus lower fitness values represented better solutions (fewer constraint violations). Table 1 shows the average fitness of 96 runs using both single crossover and double crossover GAs to solve the problem. Note that each GA was allowed to run until it converged on a single fitness value for all members of its population. While Table 1 may not show any significant advantage gained by using a double crossover, it is clear that nothing was lost by its use, as was expected.

Lastly, Figure 3 shows that the greedy crossover method proved to be an excellent choice. To implement a greedy crossover, the double crossover operation was altered to select from among a given set of intersection points the two elements which would create the shortest resulting possible solution. The result produced only one child for each parent pair, thus twice as many crossover operations were performed per generation. By adding a greedy method to the crossover, the GA tended to find equivalent path lengths 5 to 10 generations earlier than was possible without the greedy crossover. Even though the computation needed to perform a crossover increased, the overall execution of the GA was lessened by lowering the number of generations required to solve the problem.

Table1. Crossover comparison for simple scheduling problem

Crossover used	No. runs	Average fitness
single crossover	96	14.2
double crossover	96	11.0

Table2. Mutation operators and the PMX crossover

Mutation used	No. runs	Best fitness	Average fitness
no mutation	44	1266	1311.2
limited mutation	44	1240	1291.8
unlimited mutation	44	1224	1283.9

5.2 Mutation

The consequences of using the mutation operator were more difficult to characterize. Two different types of simulations were run which tested mutations with both constant and variable length strings. The first simulation, built using the PMX operator and the swapping mutation described in the latter part of section 3.1, was run on a traveling salesman problem covering 40 cities. Three types of runs were made: one with no mutation, one with a limited mutation which was allowed only outside of the cut region, and one with an unlimited mutation which could occur anywhere in the solution. All runs were allowed to converge and a total of 44 runs of each type were recorded. Table 2 shows the results of these simulations. Though the prediction of better performance using a mutation that could effect the inside of a cut region was shown to be true, its effects were not as large as expected. This may have been due, in part, to the fact that relatively small population sizes were used (30 to 50 members), but the extent to which population size has an effect upon mutation was not investigated. More statistical results will have to be presented on such data. The conclusion drawn here was that mutations which could occur inside the cut region certainly had no adverse effects upon the order-dependence of the solution strings, and may have enhanced the search for a more optimal solution.

The second simulation, in which an attempted was made to translate the benefits of mutating inside the cut region to the MT genetic algorithm, proved to be fruitless. A swapping mutation operation such as the one used with the PMX crossover could not be used since the result would, in general, produce an invalid path. In fact no single-locus mutation operator could be devised. The scheme finally adopted was to select a cut region similar to the double crossover cut by randomly picking two points of the solution string. All elements between the mutation points were removed and a new random path was generated to bridge the gap. The mutation probability also had to be changed since it now referred to the whole solution rather than each locus individually. With each solution averaging approximately 50 elements, this yielded about a 5 percent (50 x 0.001 for each locus) total chance that any given string should mutate. Figure 4 shows that the use of this mutation operator had no effect on the performance of the mobile transporter GA.

Part of the reason for this poor performance may be explained by the fact that the mutation operator chosen was not dependent upon the order of a schema. By choosing a *region* for mutation, rather than some point-by-point operation, schema which survive the mutation process will most likely be short in length. But this is exactly the situation already enforced by a double crossover, thus no further preferences result from the use of the mutation operator. Consequently, as is shown by Figure 4, nothing is gained by adding the mutation operator. However, it is important to note that this is only true for the mutation operator defined here; it may be possible to define other mutation techniques which would have a beneficial

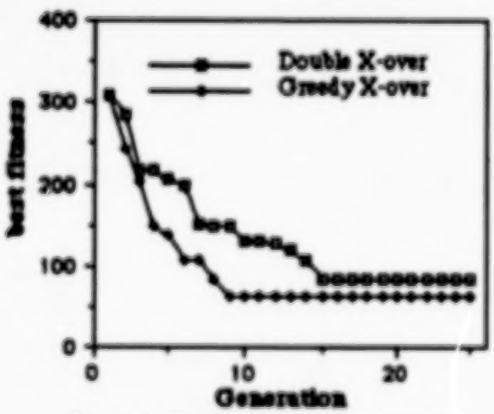


Figure 3. Double and greedy crossovers

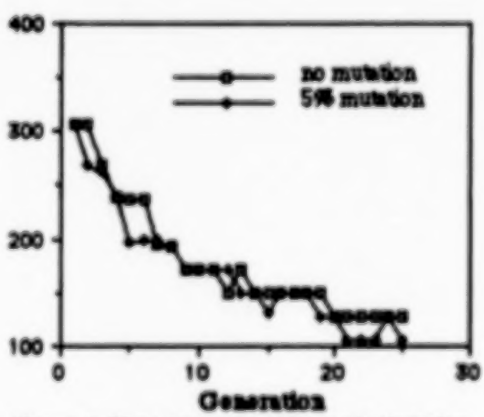


Figure 4. Mutation in the MT genetic algorithm

effect on the GA.

5.3 Local Optimization

Without an effective mutation operator which could keep the diversity of the population high, it became important to look for a means of directing the GA more quickly towards an optimal solution. Both the local optimization method we employed and the use of tournament selection helped in this regard. Local optimization was implemented by taking advantage of the inherent geometric regularities of the Space Station truss structure. Referring back to Figure 1, note that large sections of the truss structure form contiguous planes. These planes can be thought of as single units or "blocks," broken apart only when some obstacle attached to the truss structure divides the plane, or when tone plane intersects another. Rather than forming a path point by point, one can work block by block, where a block represents a contiguous length of unbroken space along some plane of the truss structure.

Figure 5 is a dramatic example of the effects of using this local optimization technique on the MT path planner problem. Not only does the average fitness of the population start out much better, but the overall performance of the GA is twice as good by the time the population converges to a single answer. In some sense this is not too surprizing since the optimization technique used greatly reduces the search space for the GA. However, it is clear that such techniques can enhance the final output of a GA when features of the problem

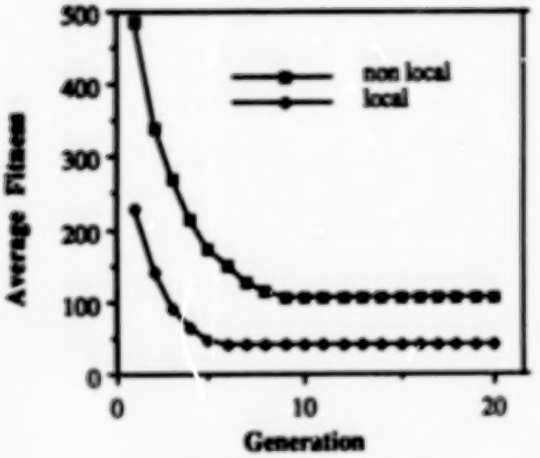


Figure 5. Effects of local optimization

domain permit their use.

Finally, Figures 6 and 7 illustrate a comparison of roulette and tournament selection methods. Tournaments of 2 were used for these experiments. As expected, tournament selection slows the rate at which a super-fit parent is allowed to proliferate through the population. The result is a higher diversity for a longer period of time, as can be clearly seen in Figure 7. Because no mutation operator could be designed for the MT path planner, this prolonged diversity became important to the overall performance of the GA. Without tournament selection, the GA converged upon a solution too quickly, missing better solutions latent in the population.

6. SUMMARY

The genetic algorithm approach proved to be an effective means for quickly solving

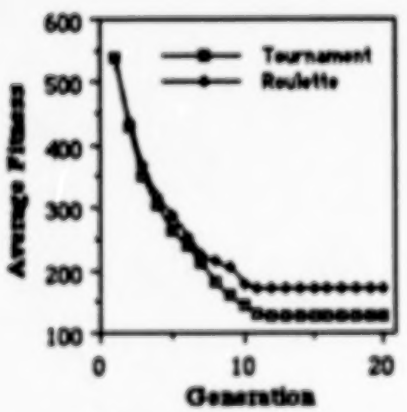


Figure 6. Tournament avg. fitness

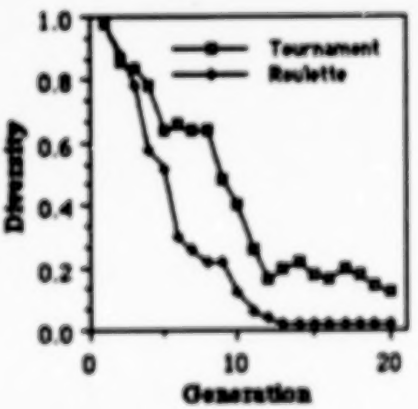


Figure 7. Tournament diversity

the mobile transporter path planning problem. Even though the MT problem did not map directly into a GA format because of the variable length of its solutions, a GA was successfully applied which produced optimal results within relatively few iterations of the algorithm. This GA used a greedy double crossover, tournament selection, local optimization which grouped contiguous points of the truss structure into "blocks," and a delooping procedure to remove cycles from offspring. The greatest improvement in performance came with the addition of local optimization and greedy crossover. Neither the presence of the delooping procedure, nor the fact that some points of the domain space were more likely to be chosen as crossover points, had any adverse effects upon the efficacy of the algorithm. No mutation operator was used since none could be devised to effect solutions on a point-by-point basis. More research still needs to be done on both the mutation operator and the effects of using an intersection set for the choice of crossover points.

Ultimately, an operational solution to the MT path planning problem may be implemented using an exhaustive coding technique due to the relatively small size of its problem domain. However, since trajectory planning is a common problem in space systems which have much larger domains, the genetic algorithm will remain an attractive alternative to the classical techniques used to solve such problems.

7. ACKNOWLEDGMENTS

The authors would like to thank Dr. David Goldberg and Robert Smith of the University of Alabama for their explanations of genetic algorithm techniques and their helpful discussions.

8. REFERENCES

1. M. A. Gomez-Tierno, Study on the use of the genetic algorithm for the solution of global optimization problems, Government Report #GMV-204/85-V2/85, 151 pages (1985).

2. J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor (1975).

3. D. E. Goldberg, "Simple genetic algorithms and the minimal, deceptive problem," in Genetic Algorithms and Simulated Annealing, Lawrence Davis, ed., Morgan Kaufmann Publishers, Inc., Los Altos, 74 - 88 (1987).

4. D. E. Goldberg and R. Lingle, "Alleles, loci, and the traveling salesman problem," in Proc. Intl. Conf. on Genetic Algorithms and their Applications, 154 - 159 (1985).

5. G. E. Liebens, M. R. Hilliard, M. Palmer, and M. Morrow, "Greedy genetics," in Proc. Intl. Conf. on Genetic Algorthims and their Applications, 90-99 (1987).

S O A R 89

Space Station

Space Suit Test Program

Nasa - Jsc

Crew & Thermal Systems Division

Prepared By:

Joseph Kosmo/ EC3

Philip West/ EC3

Presented By:

Michael Rouen/ EC3

Space Station Preprototype Space Suit TEST PROGRAM

- Test Program Background
- Test Matrix Overview
- Evaluation Plan

Test Program Background

Background:

- To accomodate Space Station Freedom budget constraints, and without incurring management risk, Project Office:
  - Deferred EMU activity at Prime (Phase C/D) Contractor
  - Asked CTSD to continue supporting development activities

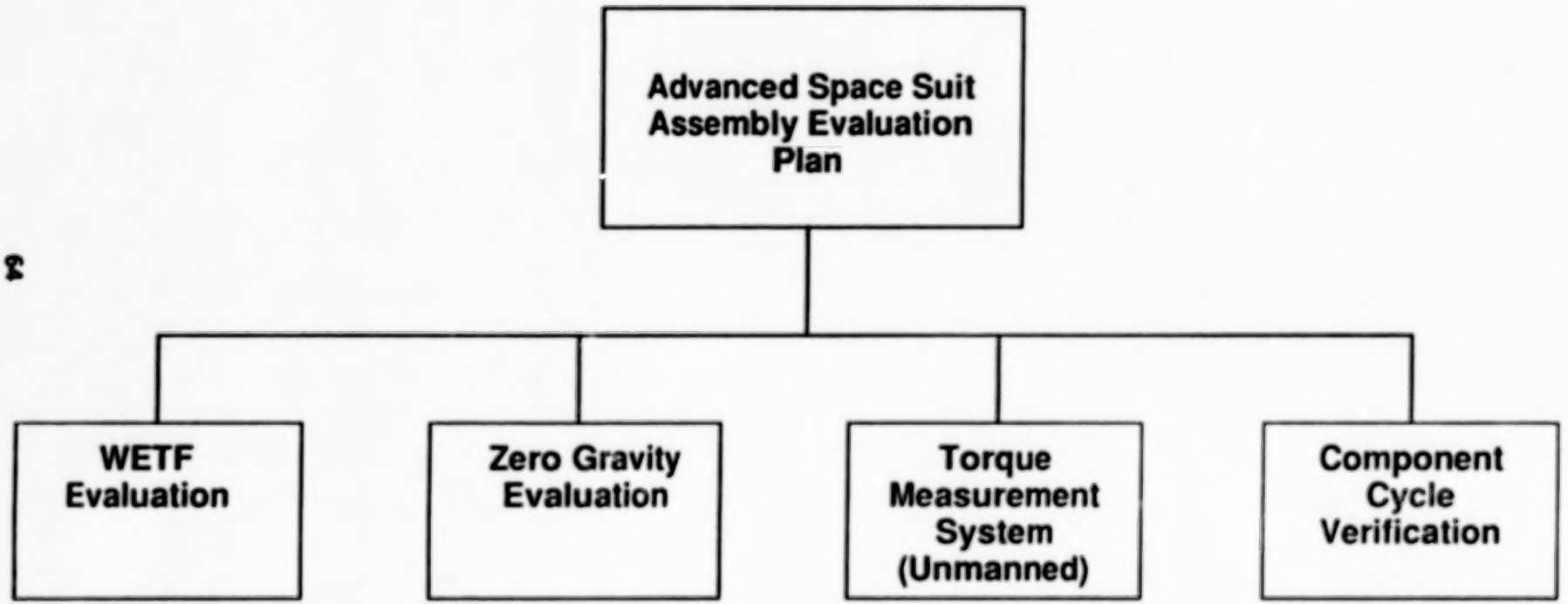
Goal:

- Develop best possible 8.3 psi space suit for Space Station FreedomProgram based on selected advanced suit technology

Objective:

- Establish quantitative measures of various performance characteristics as compared to Shuttle space suit:
  - Objective evaluations
  - Subjective evaluations
  - Typical task performance

ADVANCED SPACE SUIT ASSEMBLY EVALUATION ACTIVITIES



"Why" The Breadth of Program?

- **WETF Evaluation Activities:**
  - Simulation represents "hi-fidelity", real-time performance activities of actual EVA operations and tasks.
  - Establishes a "user" input comparison baseline developed over long-duration test excercises.
- **Zero-Gravity Evaluation (KC-135 Aircraft):**
  - Provides proper environment for don/doff evaluation activities.
  - Eliminates water inertia influencing factors.
- **Torque/Range Measurement (Unmanned)**
  - Establishes ultimate performance characteristics.
  - Provides absolute/non-subjective data-base.
- **Component Cycle Verification (Selected Joint Elements):**
  - Establishes design confidence level.
  - Identifies if design compromised due to material selection or fabrication/assembly process.

TEST MATRIX DESCRIPTION

SET-UP

PROCEDURES

REPRESENTATIVE DATA

WETF
EVALUATION

OBJECTIVE EVALUATIONS

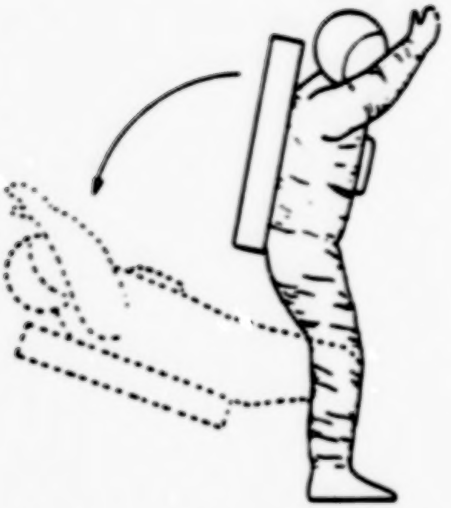
- **MOBILITY (RANGE OF MOTION)**
- **REACH ENVELOPE**
- **MAXIMUM FORCE TRANSMISSION**

SUBJECTIVE EVALUATIONS

- **MOBILITY (PERFORMANCE INDEX)**
- **EVA TASKS I**
- **EVA TASKS II**



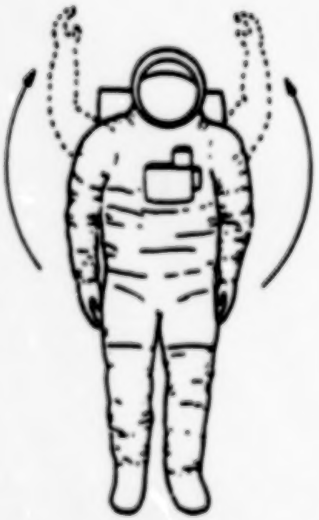
FORWARD AND UPWARD REACH FROM SIDE OF BODY (BOTH ARMS)



BACKWARD TORSO BENDING USING ANKLE, KNEE, AND TORSO



FORWARD TORSO BENDING USING ANKLE, KNEE, AND TORSO



OVERHEAD REACH FROM SIDE OF BODY (BOTH ARMS)



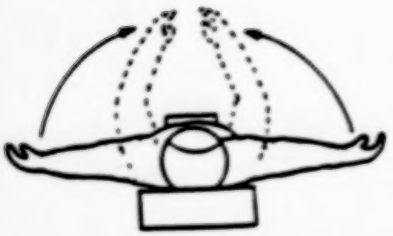
INBOARD CHEST REACH (BOTH ARMS)

BLANK

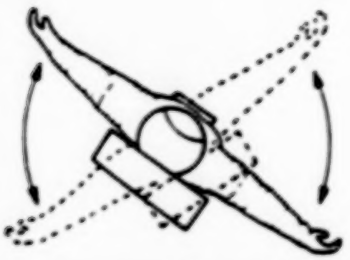
PAGE

WETF EVALUATION

ACTIVITY	OBJECTIVES
MOBILITY	Objectively evaluate SSA performance by maximum joint angle measurement during various movements Subjectively evaluate SSA using performance index throughout motion Familiarize crewmember with SSA while performing isolated joint motions
REACH ENVELOPE	Objectively evaluate SSA by defining shape and volume of one and two handed functional reach envelopes Familiarize crewmember with integrated mobility of SSA
MAXIMUM FORCE TRANSMISSION	Objectively evaluate SSA by measuring maximum force transmission for movements 1) Frequently used during EVA 2) Defined for joint isolation Familiarize crewmember with SSA mobility under heavy work loads



ARM SWEEPING MOTIONS (RIGHT TO LEFT, LEFT TO RIGHT)



TORSO ROTATION (ARMS EXTENDED)



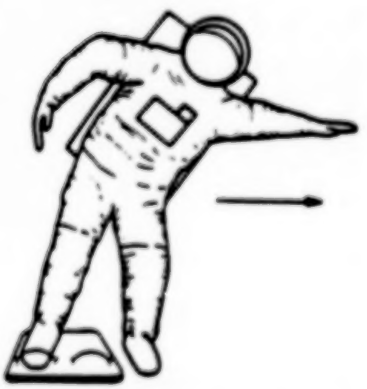
FORWARD TORSO BENDING WITH TORSO ROTATED



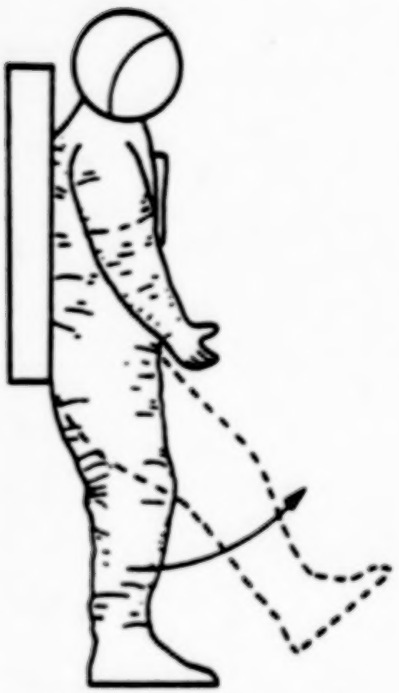
SIDE-TO-SIDE ANKLE FLEXION / REACH (RIGHT AND LEFT, BOTH FEET IN FOOT RESTRAINTS)



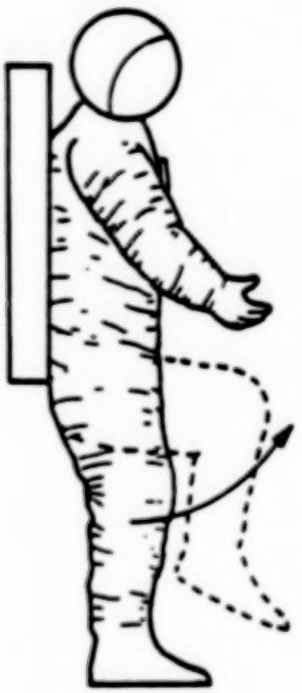
SIDE-TO-SIDE ANKLE FLEXION / REACH (RIGHT AND LEFT, ONE FOOT IN FOOT RESTRAINT)



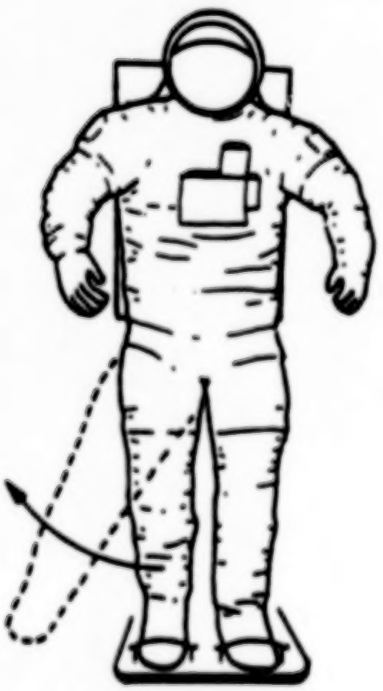
SIDE-TO-SIDE ANKLE FLEXION / REACH (RIGHT AND LEFT, OTHER FOOT IN FOOT RESTRAINT)



STRAIGHT LEG HIP FLEXION
(BOTH LEGS)

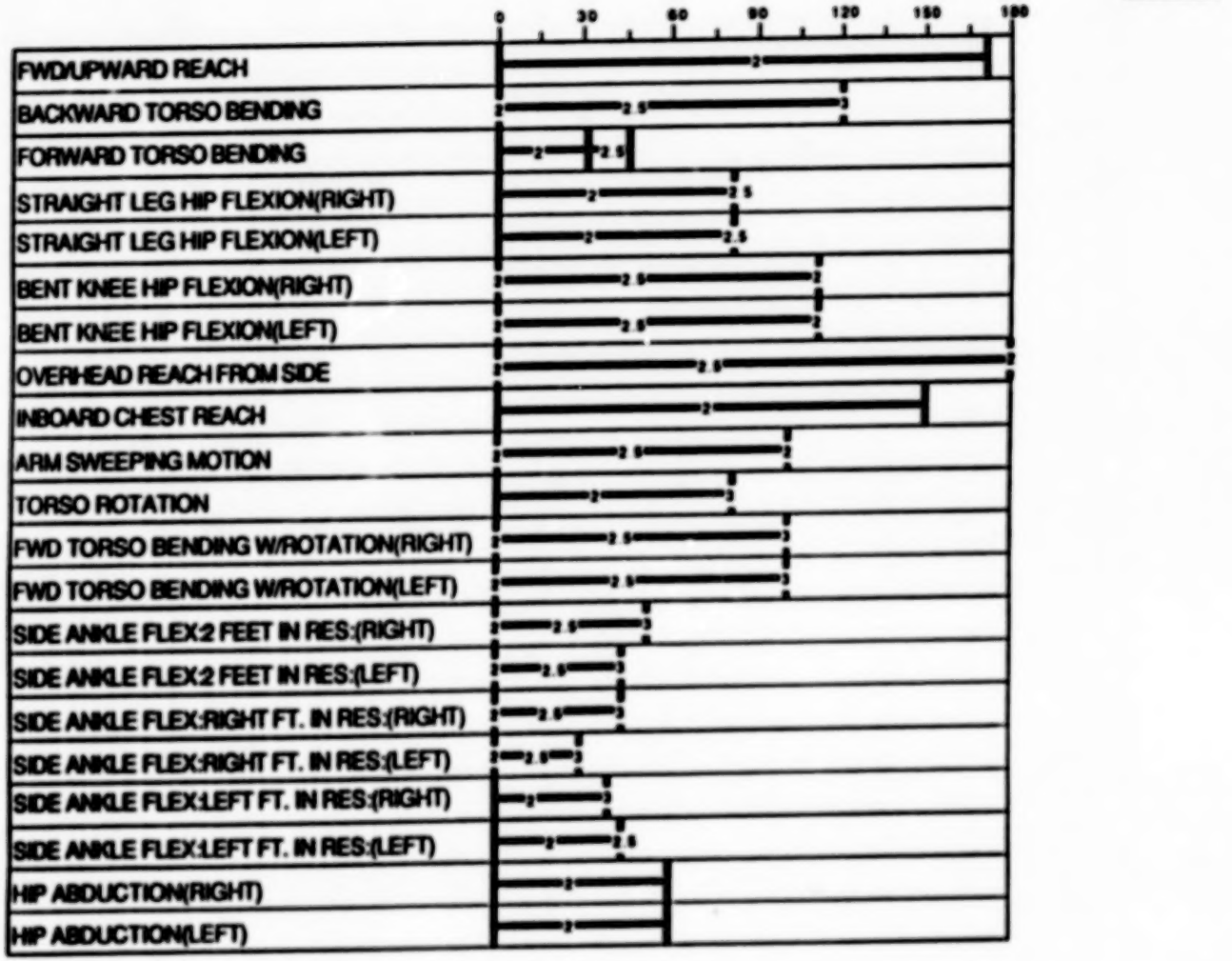


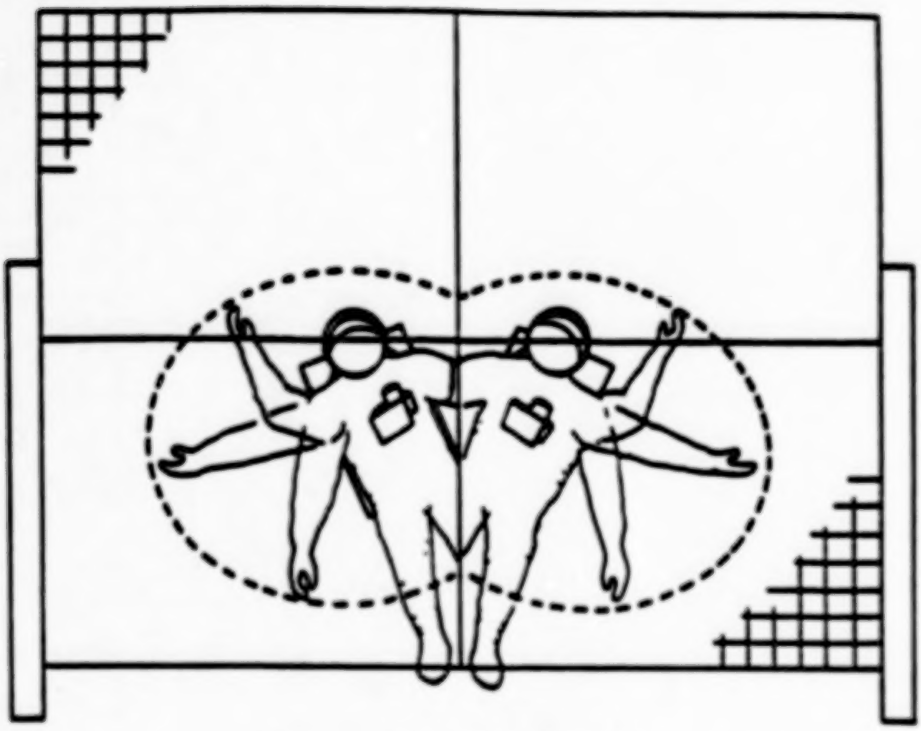
BENT KNEE HIP FLEXION
(BOTH LEGS)



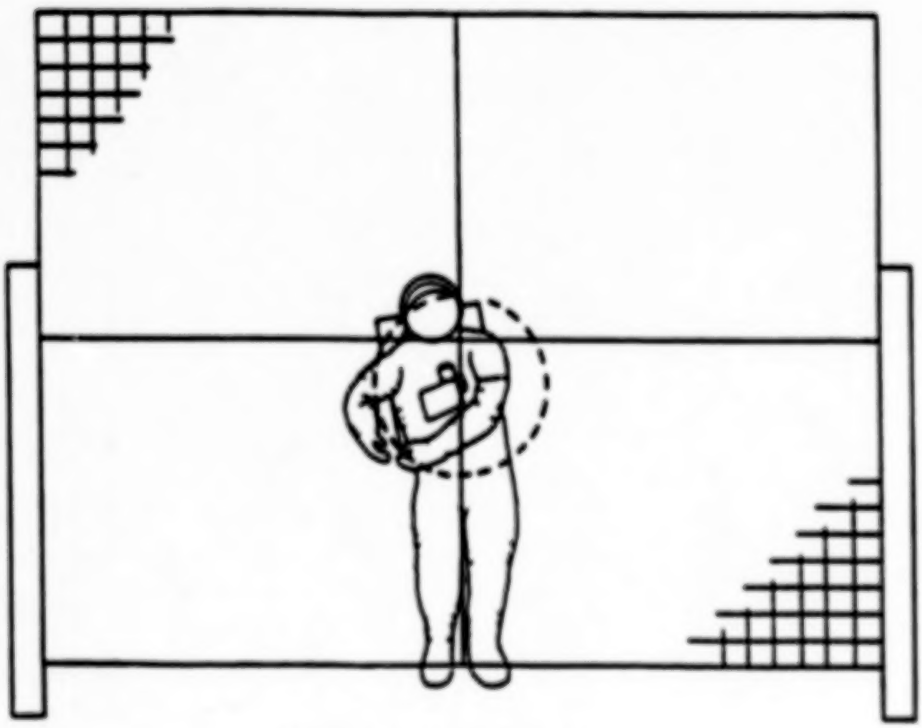
HIP ABDUCTION (BOTH LEGS)

SUIT MOBILITY EVALUATION

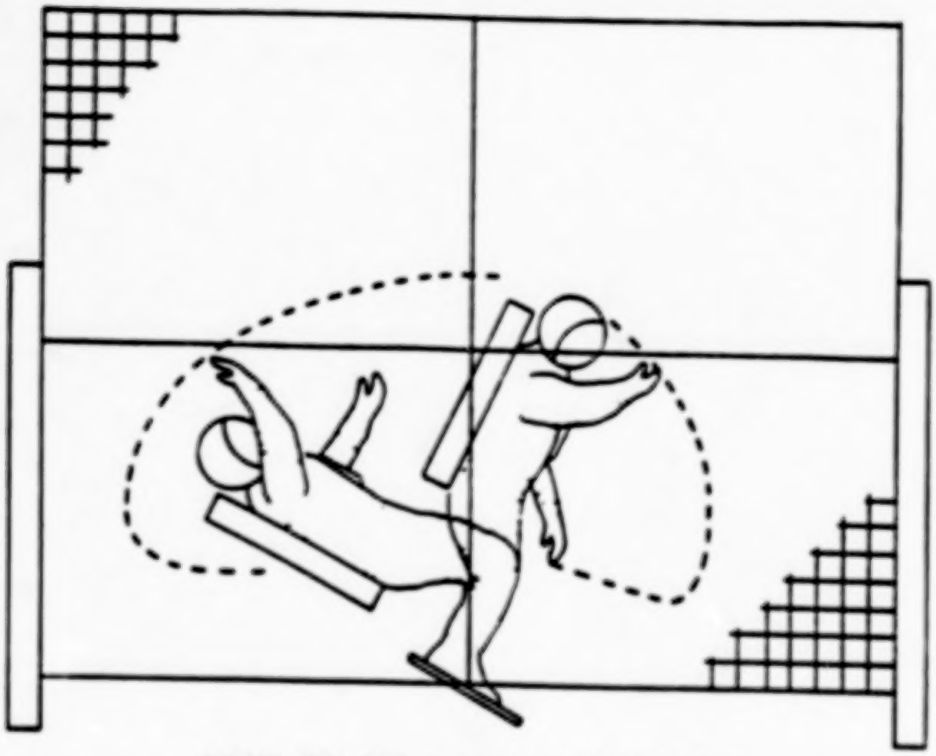




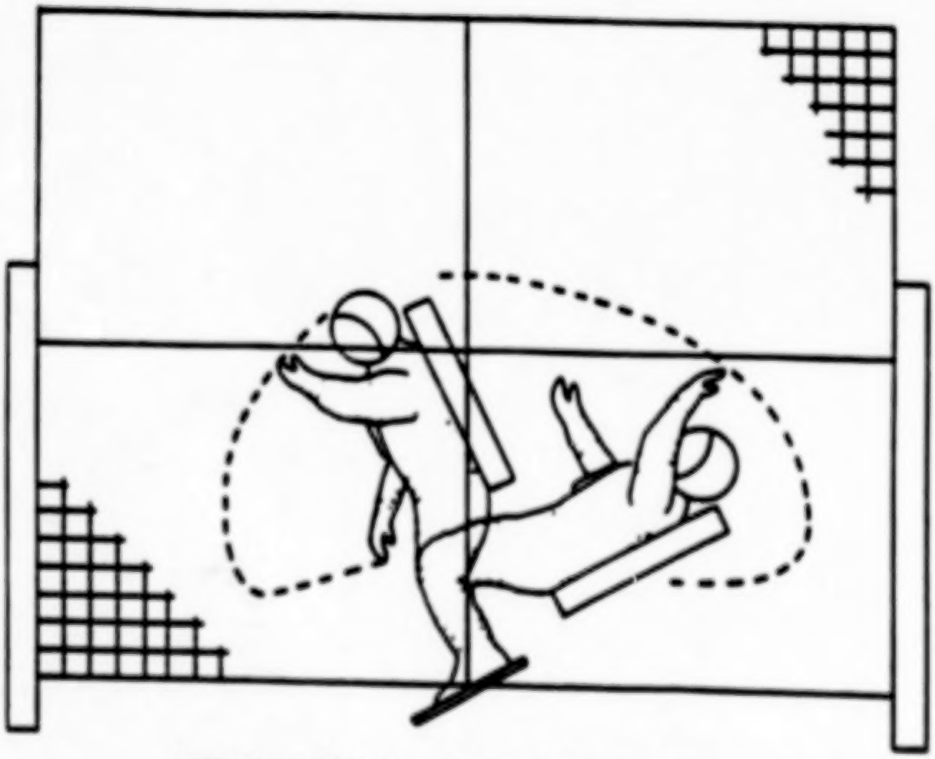
FRONT RIGHT- AND LEFT-HAND
REACH ENVELOPE



TWO-HAND REACH ENVELOPE



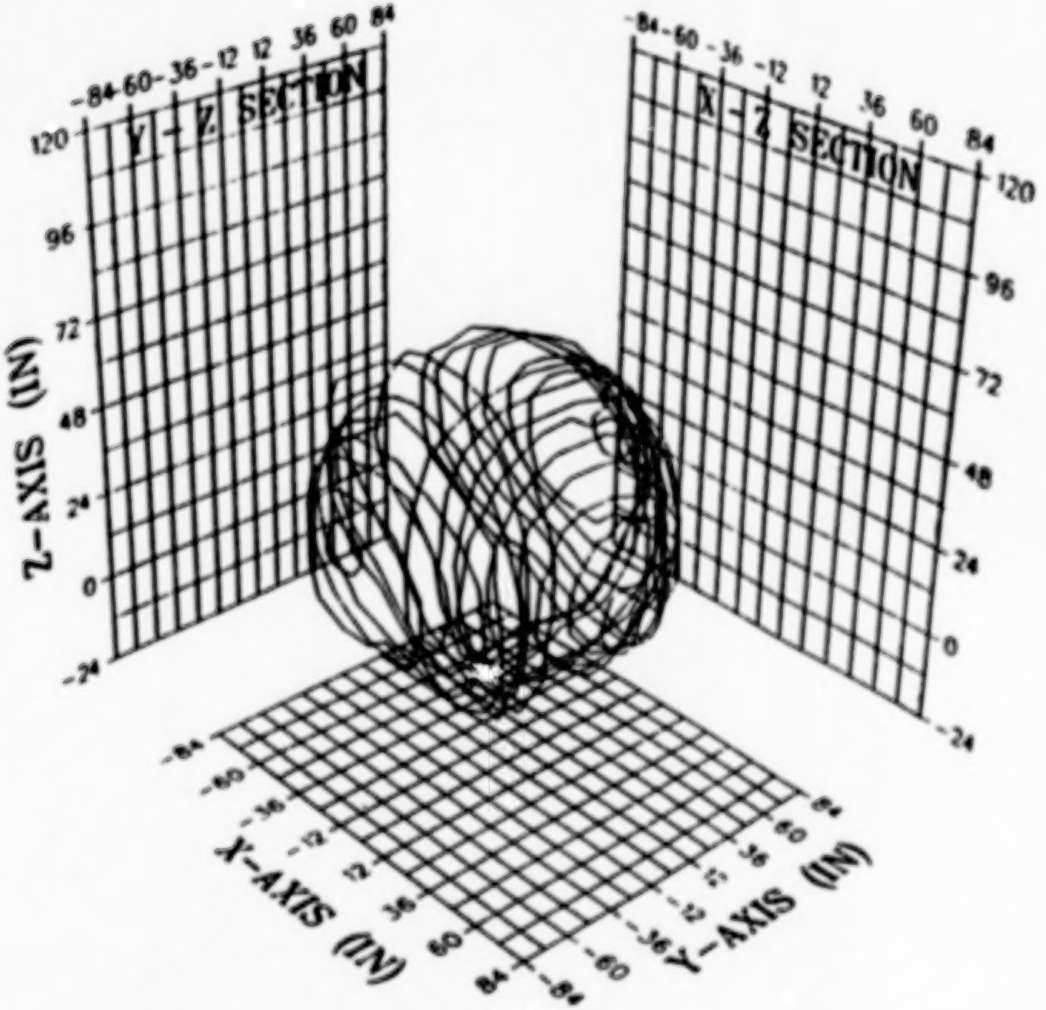
RIGHT SIDE ONE-HAND REACH ENVELOPE



LEFT SIDE ONE-HAND REACH ENVELOPE

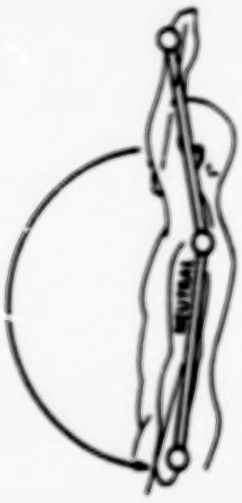
FUNCTIONAL REACH ENVELOPE

ONE-HANDED





SHOULDER ABDUCTION



SHOULDER FLEXION

SHOULDER FLEXION/ABDUCTION
(combination of first two)



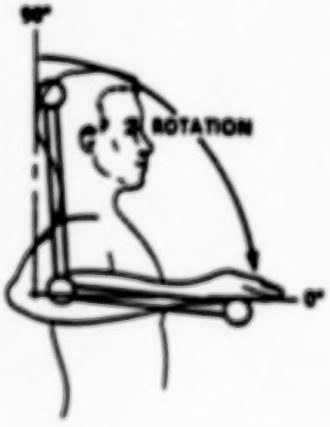
EVA RATCHET TOOL
CRANK



EVA RATCHET TOOL
PUSH / PULL



ELBOW
FLEXION / EXTENSION



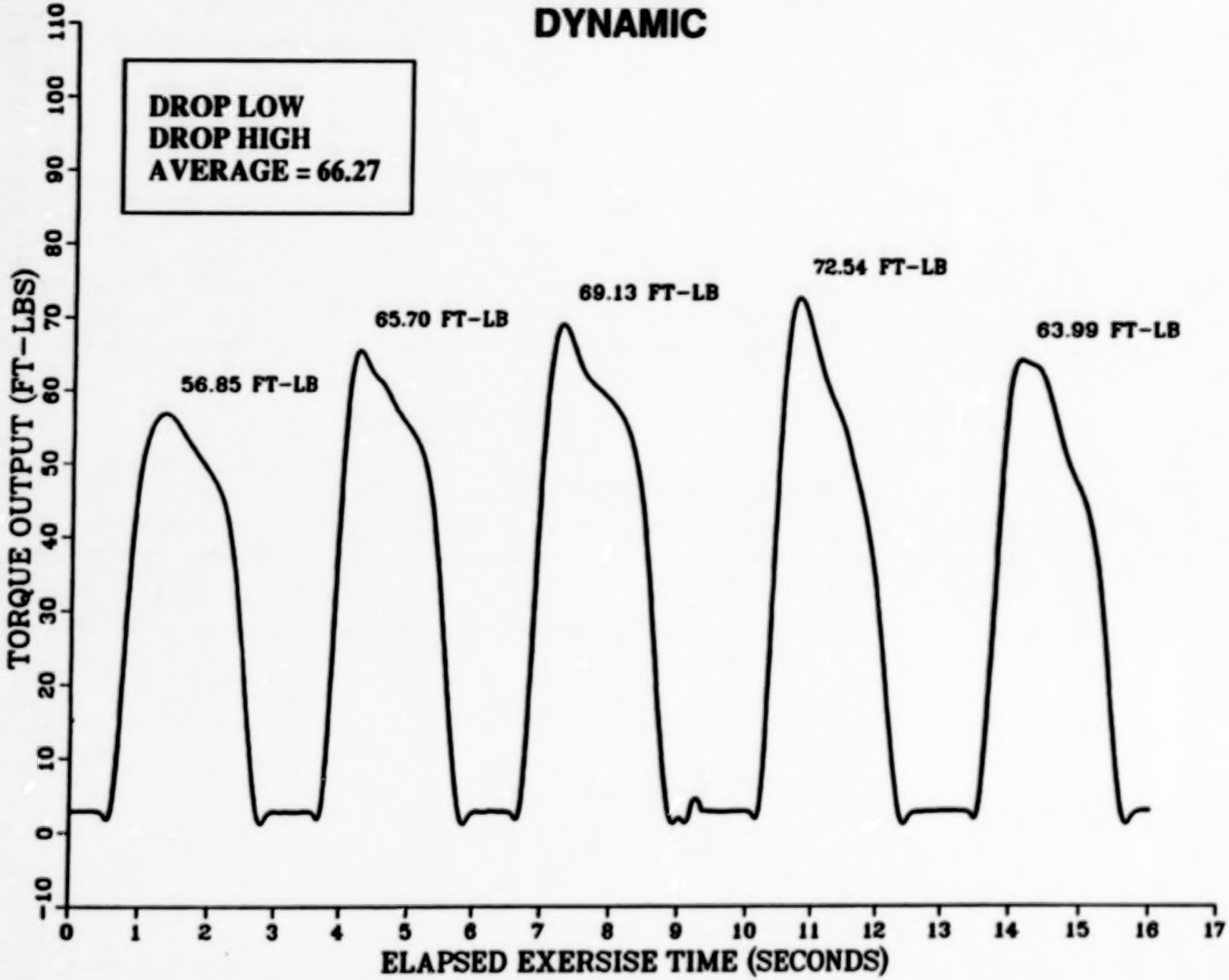
SHOULDER ROTATION
Y - AXIS



SHOULDER ROTATION
MEDIAL INTERNAL

SHOULDER FLEXION/ABDUCTION

DYNAMIC

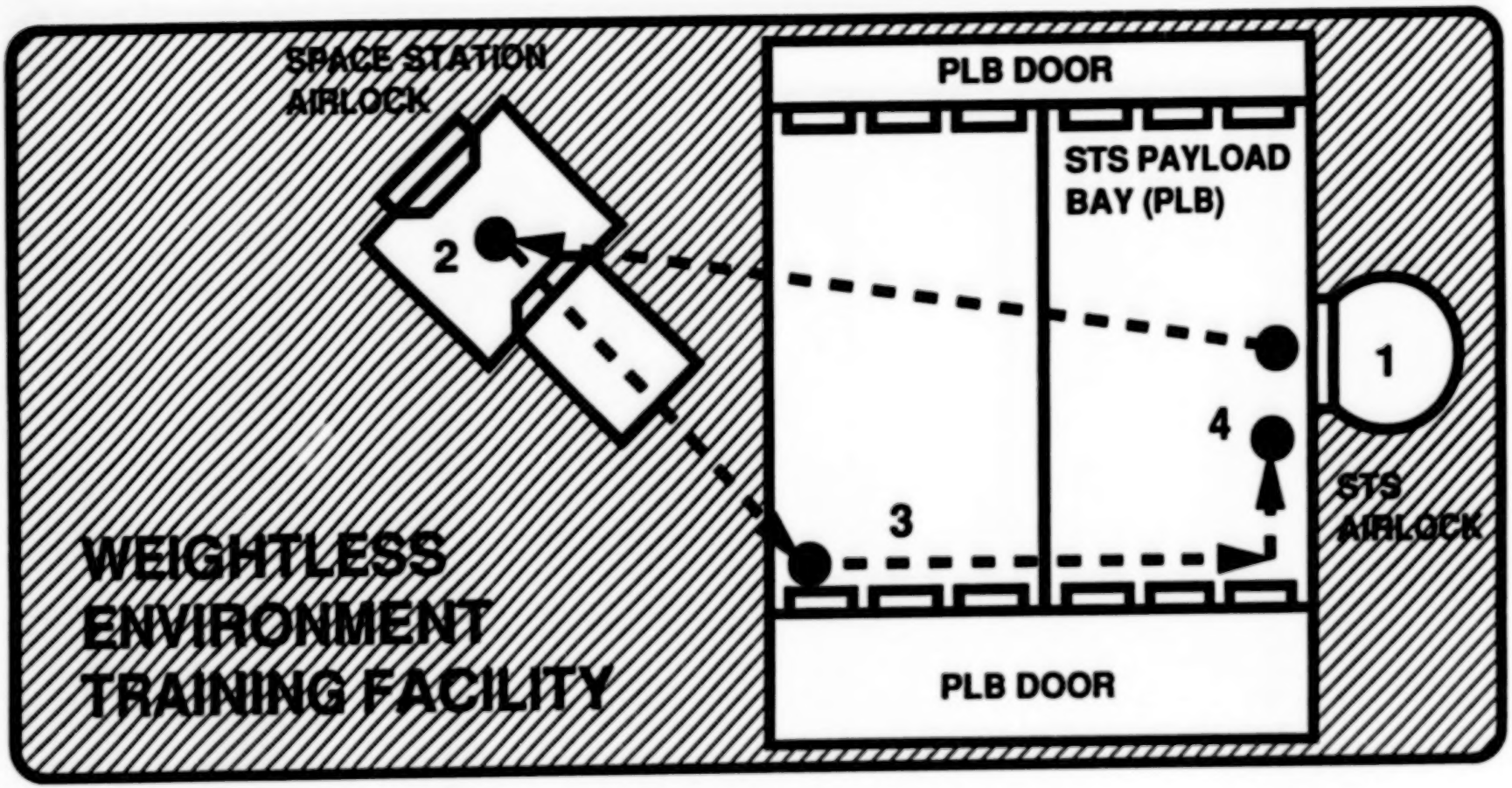


77

WETF EVALUATION
(CONTINUED)

ACTIVITY	OBJECTIVES
EVA TASKS I	Subjectively evaluate SSA using Cooper-Harper rating scale while performing common EVA tasks Familiarize crewmember with SSA mobility as used on practical applications - precursor for EVA tasks II
EVA TASKS II (EASE/ACCESS)	Subjectively evaluate SSA using Cooper-Harper rating scale while performing EASE/ACCESS assemblies and disassemblies - best representation of unrestricted complex movements while performing typical Space Station assembly tasks

EVA TASKS I



1 STS AIRLOCK OPS
2 SPACE STATION AIRLOCK OPS
3 STS PAYLOAD BAY TRANSLATION
4 STS CONTINGENCY EVA OPS
WINCH
THREE POINT TOOL

KC-135
EVALUATION

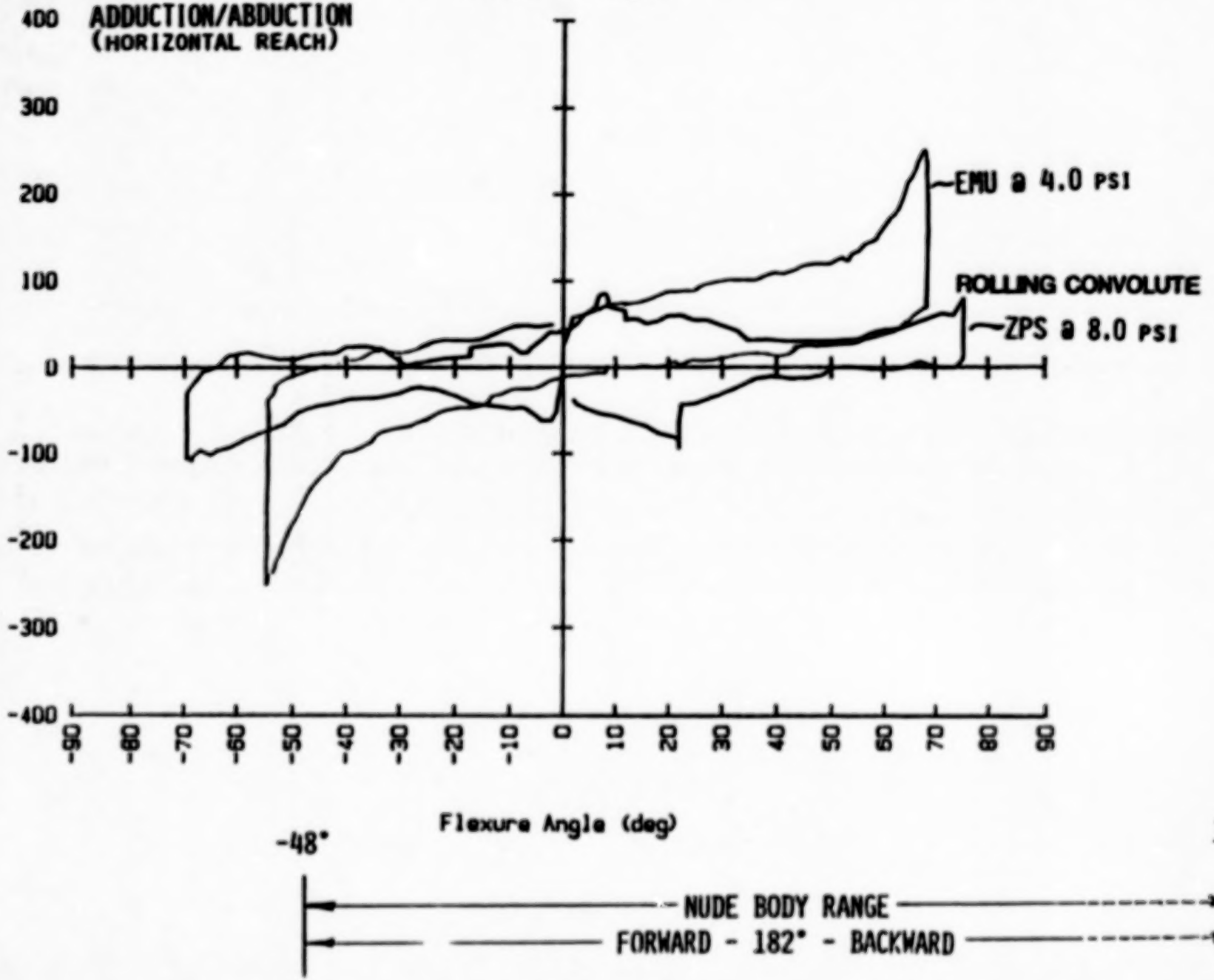
ACTIVITY	OBJECTIVES
DON/DOFF	Subjectively evaluate SSA don/doff operations using Cooper-Harper rating scale
TRANSLATION	Subjectively evaluate differences in SSA performance between neutral bouyancy (WETF) and zero - g - Ease of operation - Fit - Comfort

TORQUE / RANGE MEASUREMENT EVALUATION

ACTIVITY	OBJECTIVES
TORQUE VERSUS RANGE OF MOTION MAPPING	Objectively determine 1) Torque required to move the joint through a given range of motion 2) Maximum joint range of motion

SHOULDER JOINT COMPARISONS

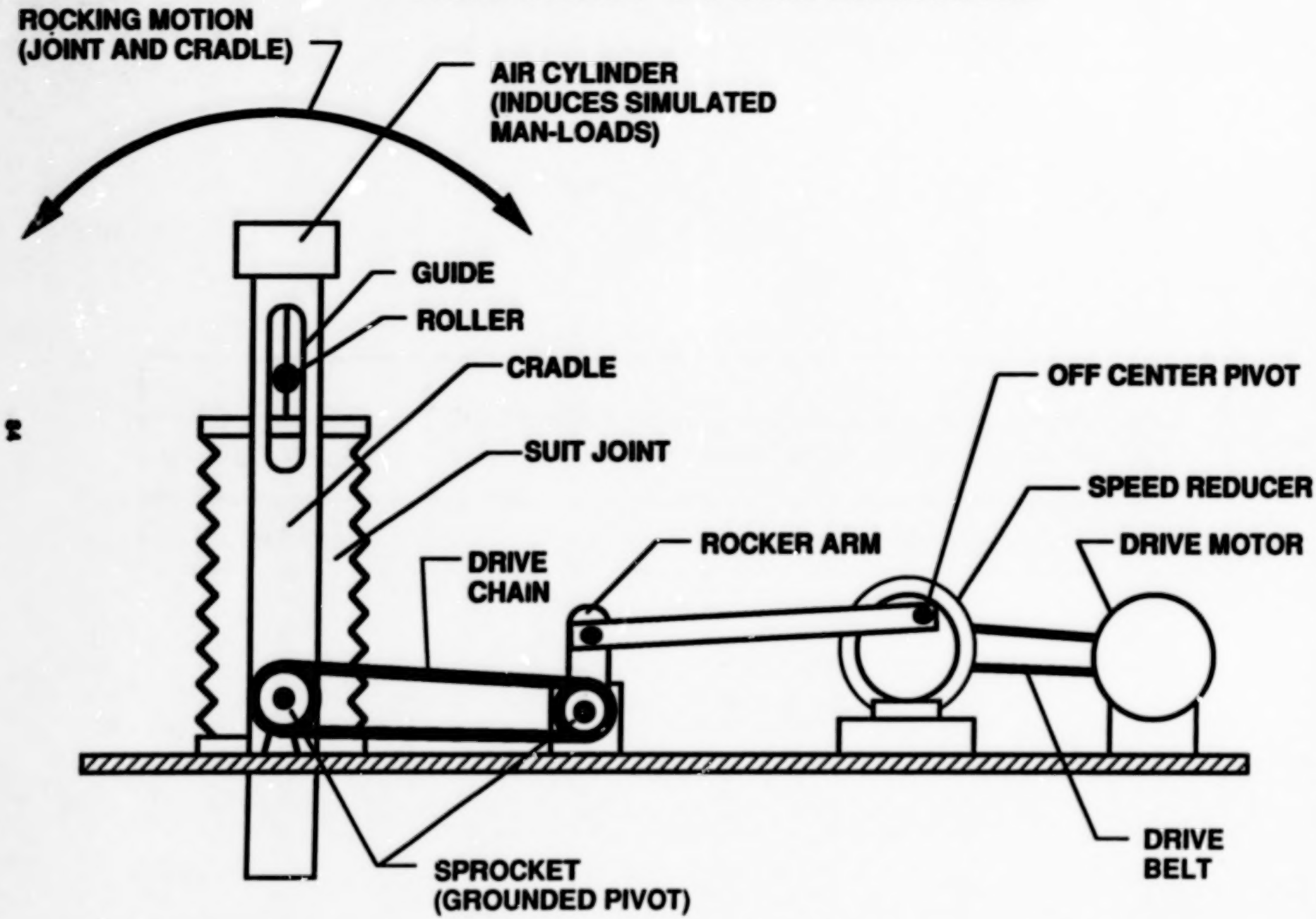
(Torque vs. Angle)



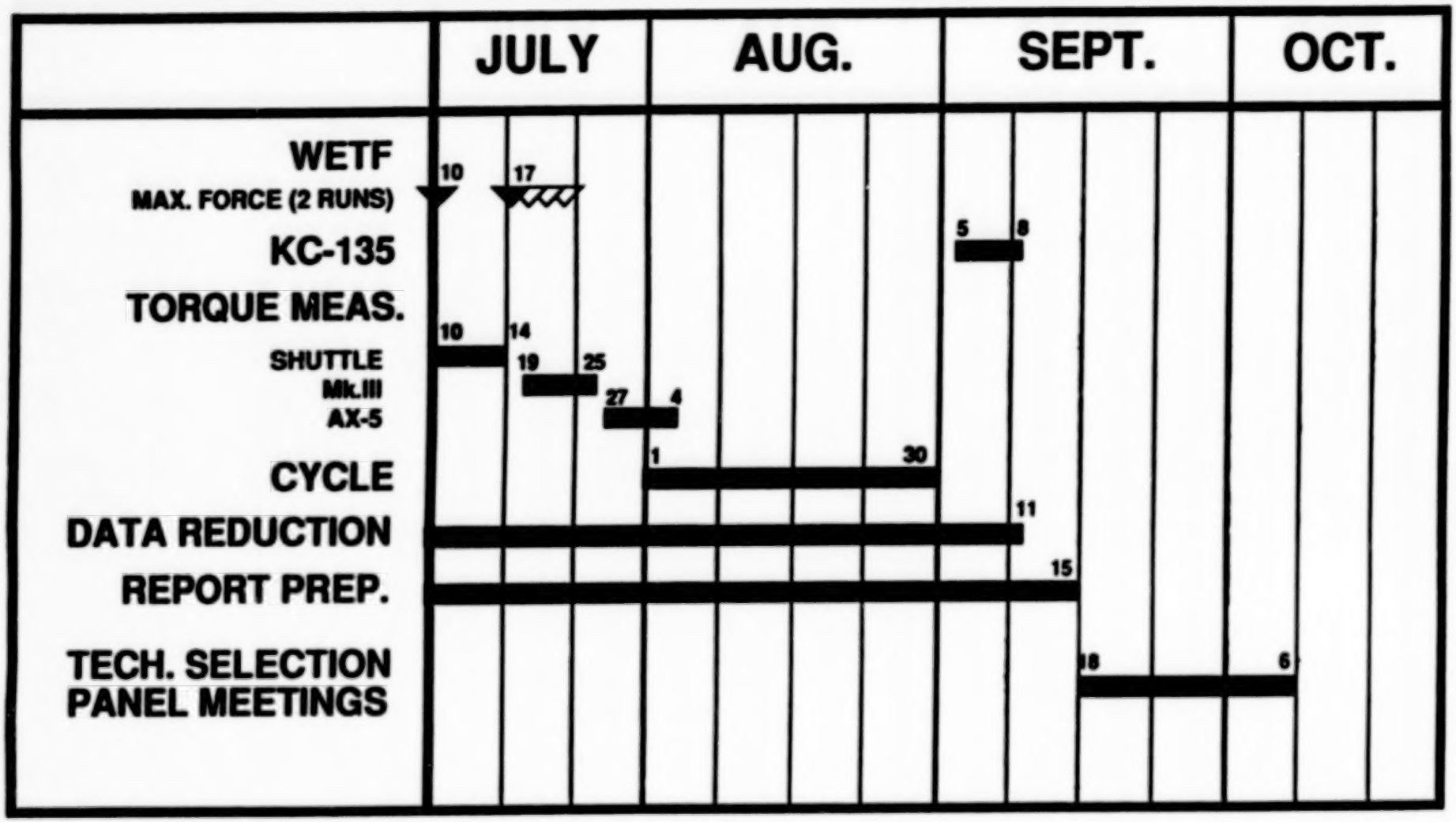
CYCLE VERIFICATION EVALUATION

ACTIVITY	OBJECTIVES
CYCLE JOINTS	Verify joint operational capability for one year on orbit life (plus a safety factor of two and based on 52 eva's per year)

SSA JOINT CYCLE MACHINE



SUIT TEST PROGRAM SCHEDULE



TASK COMPLETION MATRIX

MANNED TEST ACTIVITIES

	CREWMEMBER A			CREWMEMBER B			CREWMEMBER C			CREWMEMBER D		
WETF	STS	AX-5	Mk. III	STS	AX-5	Mk. III	STS	AX-5	Mk. III	STS	AX-5	Mk. III
SUIT MOBILITY												
REACH ENVELOPE												
MAX. FORCE												
EVA TASKS 1												
EVA TASKS 2												
KC-135	AX-5		Mk. III	AX-5		Mk. III	AX-5		Mk. III	AX-5		Mk. III

EVALUATION PLAN

EVALUATION PLAN COORDINATION MEETINGS

- ARC
- JSC
- Wk. Pkg. II Phase C/D Contractor (McDAC / LOCKHEED)

SELECTION CRITERIA PRIORITIES

TECHNOLOGY SELECTION PANEL

SELECTION PROCESS

SELECTION CRITERIA PRIORITIES

FIRST ORDER SELECTION CRITERIA MANNED PERFORMANCE

OBJECTIVE
- MOBILITY (RANGE OF MOTION)
- REACH ENVELOPE
- MAX. FORCE TRANSMISSION

SUBJECTIVE
- EVA TASKS I & II
- MOBILITY (PERFORMANCE INDEX)
- ZERO - G

SECOND ORDER SELECTION CRITERIA ENGINEERING TEST AND ANALYSIS

- TORQUE MEASUREMENT
- CYCLE VERIFICATION
- ENVIRONMENTAL PROTECTION

THIRD ORDER SELECTION CRITERIA PROGRAMMATIC ISSUES

- LIFE CYCLE COSTS

TECHNOLOGY SELECTION PANEL

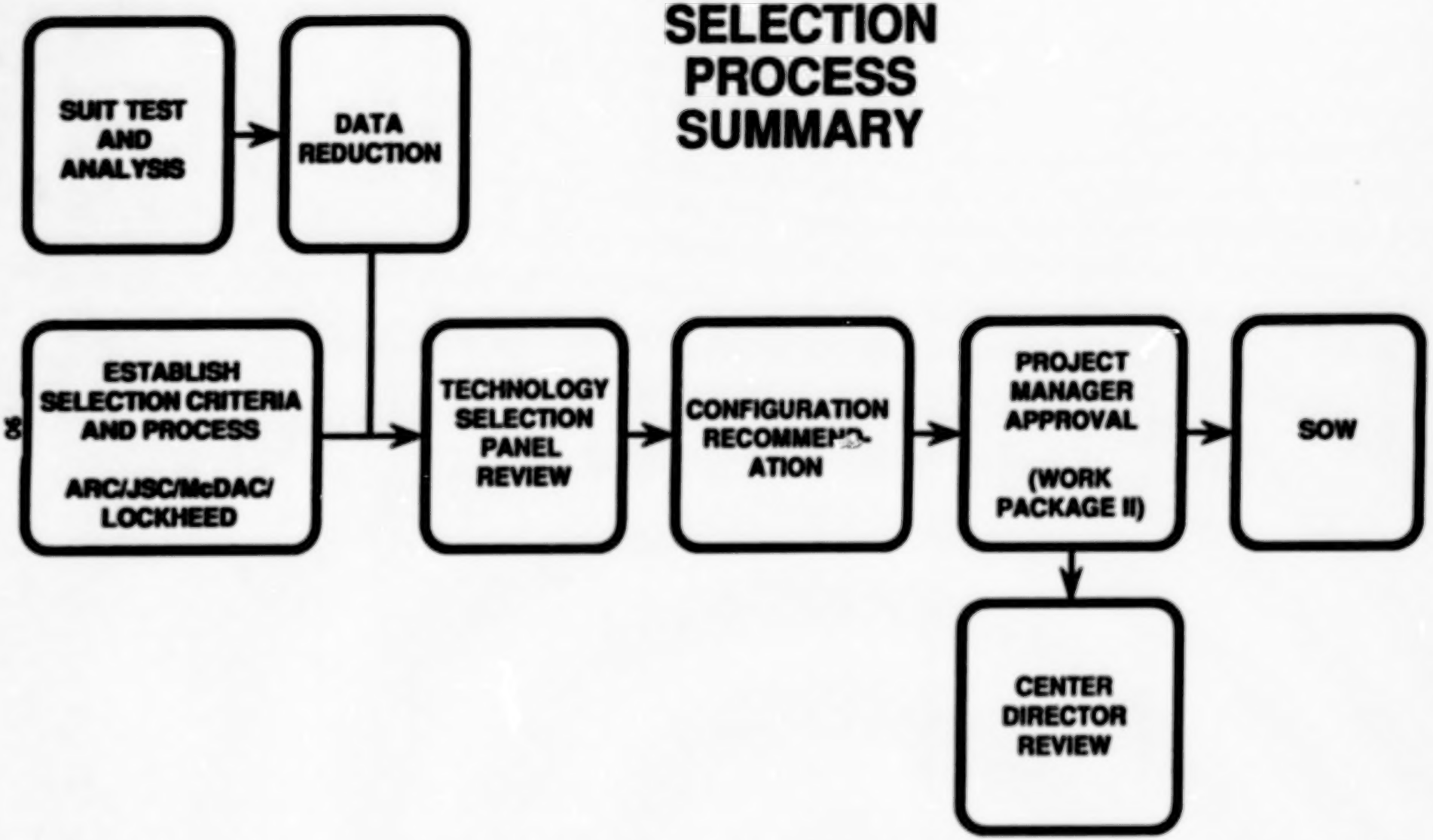
PURPOSE

- REVIEW ALL TEST DATA
- MAKE TECHNOLOGY SELECTION RECOMMENDATION

MEMBERSHIP

- CHAIR: EMU SYSTEM DEVELOPMENT MANAGER (SDM)
  - ROUEN
- TECHNICAL EXPERTS
  - KOSMO/JSC
  - VYKUKAL/ARC
- ASTRONAUT OFFICE
  - ROSS
- SPACE STATION PROJECT OFFICE (Wk. Pkg. II)
  - KISSINGER
- SYSTEMS ENGINEER
  - WEBBON/ARC
  - WEST/JSC
- PHASE C/D CONTRACTOR (Wk. Pkg. II)
  - RAFFAELLI/Mc DONNELL DOUGLAS
  - WILKINSON/LOCKHEED

TECHNOLOGY SELECTION PROCESS SUMMARY



AX-5 ADVANCED SPACE SUIT DESIGN OVERVIEW

Captain A. Reinhardt
NASA/Ames Research Center

(Paper not provided by publication date.)

COMBAT EDGE: ADVANCES IN TACTICAL FIGHTER PRESSURE GARMENTS

Major T. Morgan
Brooks AFB

(Paper not provided by publication date.)

MULTISYSTEM APPROACH TO MANNED DIVING OPERATION

J. Donnelly
Ocean Systems Engineering, Inc.

(Paper not provided by publication date.)

SPACECRAFT CONTAMINATION PROGRAMS WITHIN THE AIR FORCE SYSTEMS COMMAND LABORATORIES

Edmond Murad
Geophysics Laboratory (AFSC), Hanscom AFB, MA, 01731

ABSTRACT

Spacecraft contamination programs exist in five independent AFSC organizations: Geophysics Laboratory (Lead Laboratory), Arnold Engineering & Development Center (AEDC), Rome Air Development Center (RADC/OCSE), Wright Research & Development Center (MLBT), Armament Laboratory (ATL/SAI), and Space Systems Division (SSD/OL-AW). In addition, a sizable program exist at Aerospace Corp., an FCRC for AFSC. These programs are complementary, each effort addressing a specific area of expertise: GL's effort is aimed at addressing the effects of *on-orbit* contamination; AEDC's effort is aimed at ground simulation and measurement of optical contamination; RADC's effort addresses the accumulation, measurement, and removal of contamination on large optics; MLBT's effort is aimed at understanding the effect of contamination on materials; ATL's effort is aimed at understanding the effect of plume contamination on systems; SSD's effort is confined to the integration of some contamination experiments sponsored by SSD/CLT; and Aerospace Corp.'s effort is aimed at supporting the needs of the using SPO's in specific areas, such as contamination during ground handling, ascent phase, laboratory measurements aimed at understanding on-orbit contamination, and mass loss and mass gain in on-orbit operations. These programs will be described is some detail, with emphasis on GL's program.

INTRODUCTION

Systems operating in low earth orbit, have experienced anomalies and artifacts in their observations. Some of the anomalies are attributable directly to contamination in and near the spacecraft surfaces. An example of contamination effects is the observation of gaseous contaminants in the shuttle bay which are due to the return flux of gases from thrusters fired at various locations in the Space Shuttle (Narcisi et al., 1983; Wulf and von Zahn, 1986) and the observation of enhanced ionization and suprathermal ions in the vicinity of the shuttle (Hunton and Calo, 1985; Grebowsky et al., 1987). In addition to these observations, others have been made of a closely related subject, namely the erosion of materials by atomic oxygen due to the high velocity of the ramming atmosphere (Leger and Visentine, 1986; Gregory, 1987; Leger et al., 1987; Peters et al., 1986; Peters et al., 1988). A third, and somewhat startling, contamination effect has been the report of a glow on shuttle surfaces in the ram direction (Mende et al., 1983, 1985, 1987). This latter observation has been attributed to off-surface recombination of impinging O atoms and surface adsorbed NO (Swenson,et al 1985; Swenson, 1986; Kofsky and Barrett, 1986).

DISCUSSION

These observations and reports of anomalies on a number of flights have prompted AFSC to undertake a number of different, but complementary, program to address the issue of spacecraft contamination. Listed below are the efforts in the individual laboratories and Aerospace Corp.

Arnold Engineering and Development Center (AEDC/DOTR): The focus of the work is the laboratory simulation of spacecraft contamination and the measurement of various aspects of the contaminants, such as degradataion of optical surfaces due to contamination, flowfields of plume contamination, and formation of droplets in the expansion of vapor and liquid in vacuum. The experiments are carried out in a test chamber at AEDC. Focal point for the work is AEDC/DOTR (Capt S. D. Shepherd/615-454-6517). Technical support is provided by Calspan (Mr. Bobby Wood).

Armament Laboratory (ATL/SAI): The stress of this work is on the effect of contamination generated by the return flux of plumes. As part of the work, ATL/SAI has picked up the development and support for **CONTAM 3.xx**. Focal point is Ms. Vicki Cox (ATL/SAI, 904-882-4278). Work on **CONTAM 3.xx** is carried out under contract by Dr. Ron Hoffman of Science Applications International Corp.

Rome Air Development Center (RADC/OCSE): The focus of this work is development of techniques for the measurement, mitigation, and removal of contamination from large optics. In addition, space and laboratory experiments are developed to validate these techniques. Focal point for the work is Capt Carol Moreland/315-330-3145). Technical support is provided by W. J. Schaeffer Associates (Dr. Keith Shillito).

Wright Research & Development Center (WRDC/ MLBT): The focus of this effort is the development of a database for spacecraft contamination as well as the measurement of contamination effects on materials. Focal point for the effort is Lt Arthur Estavillo (WRDC/MLBT/513-255-9022). Some technical support is provided by Aerospace Corp.

Aerospace Corp.: Aerospace Corp. has a large program to address ground-based contamination of spacecraft payloads, ascent phase spacecraft contamination, and some effects of on-orbit contamination. The work includes theoretical studies, laboratory work on fundamental rate processes, and space experiments. The focal point for the work is Dr. Graham Arnold (Aerospace Corp./MS M2-271/213-336-1935).

Geophysics Laboratory (GL/PHK): The work at GL is concerned mostly with effects of on-orbit contamination. Focal point for the work is at GL/PHK (Dr. Edmond Murad/617-377-3176). Below is a somewhat detailed description of the various parts of the effort:

Shuttle Glow: This work is aimed at understanding the nature and the cause of the Shuttle Glow phenomenon. Towards that end GL/PHK has undertaken a program for the development of high resolution spectrographs which will yield spectra over the range 115-1100 nanometers with a resolution of 0.35-1 nanometers, depending on the region. Two such instruments have been built; one will fly as part of the STS-39 mission, and the other is built for flight as a hitchhiker instrument. In addition, a hand-held camera responsive in the visible and operated by astronauts from the astronauts' bay has been flown. A new version of the latter camera will provide spectra in the visible region of the spectrum at a resolution of 0.9 nanoameters.

Particulates: An experiment to obtain photographs using a stereo camera system with a strobe light was flown. One of the cameras did not work so that only monographic pictures were obtained. Nonetheless, it was possible to obtain great deal of information about the time history of particulates in the shuttle environment, effect of sunlight and operations (e.g. satellite deployment) on the particulate environment of the space shuttle. This work was published (Green et al.,1987). As part of this work we recently conducted a dedicated water release experiment over the AMOS Observatory at Mt. Haleakala, HI. In this case we observed the formation of dense cloud of ice particles which extended for about 1.6 km and which had a width of about 0.6 km. Evaporation seemd to be a major loss mechanism. A manuscript covering this work is in preparation.

Gaseous Contamination: Some measurements of the composition of the gaseous cloud in the bay of the Space Shuttle have been performed using a quadrupole mass spectrometer. The work is carried out in the Ionospheric Physics Division (Dr. D. Hunton, GL/LID, 617-377-3048). Some of this work has been published (Narcisi et al., 1983; Hunton and Calo, 1985). In addition plans are underway to use the imager/spectrograph described above for the measurement of composition using optical techniques.

Laboratory Experiments: Work is underway to develop a source of ground state O atoms whose energy can be varied from 2 eV to 10 eV. This work is being done at the Jet Propulsion Laboratory by Drs. Ara Chutjian and Otto Orient. In this work surface recombination as it pertains to the Shuttle glow will be studied as well as neutral-neutral gas phase chemistry. In addition to the reactions of neutral beams, a double mass spectrometer at GL is being used to measure reaction cross sections and energy distributions of ions and neutrals as a function of energy. The work is being conducted by Drs. Rainer Dressler and James Gardner.

Model Development: The end result of GL's Spacecraft Contamination Program is the development of a user-friendly *Monte Carlo* code which can be used by designers to predict the effect of contamination on the operation of experiments and systems in low earth orbit. The code which we are developing with Spectral Sciences, Inc., SOCRATES, has been used to predict the optical emissions in the environment of the space shuttle, as a vernier thruster is fired. Currently SOCRATES includes scattering and some chemical reactions. As data from the laboratory measurements become available, they will be incorporated into the code. Currently under development is the second module of SOCRATES, the gas-surface interactions model. In a third, and final, module of the code we intend to incorporate ion-neutral collisions. The main developer of the code is Dr. J. Elgin of Spectral Sciences, Inc.

REFERENCES

Grebowsky, J. M., Taylor, Jr., H. A., Pharo, III, M. W., and Reese, N., Thermal Ion Perturbation Observed in the Vicinity of the Space Shuttle, *Planet. Space Sci., 35*, 501 513, 1987.

Grebowsky, J. M., Taylor, Jr., H. A., Pharo, III, M. W., and Reese, N., Thermal Ion Complexities Observed within the Spacelab 2 Bay, *Planet. Space Sci., 35*, 1463-1469, 1987.

Green, B.D., G.K. Yates, M. Ahmadjian, and H. Miranda, The Particulate Environment Around the Shuttle as Determined by the PACS Experiment, *SPIE Vol.777-Optical System Contamination:Effects, Measurement, Control*, 1987.

Gregory, J.C., Interaction of Hyperthermal Atoms on Surfaces in Orbit: The University of Alabama Experiment, in *Proc. of the NASA Workshop on Atomic Oxygen Effects*, D.E. Brinza, Editor. JPL Publication 87-14, p. 29, 1987.

Hunton, D. E., and J. M. Calo, Low Energy Ions in the Shuttle Environment: Evidence for Strong Ambient-Contaminant Interactions, *Planet. Space Sci., 33*, 945-951, 1985.

Kofsky, I.L., and Barrett, J.L., Spacecraft Surface Glows, *Nucl. Instr. Methods in Phys. Res., B14*, 480, 1986.

Leger, L. J. and J. T. Visentine, A Consideration of Atomic Oxygen Interactions with the Space Station, *J. Spacecraft Rockets, 23*, 505, 1986.

Leger, L. J., B. Santos-Mason, J. Visentine, and J. Kumlesez, Review of LEO Flight Experiments, in *Proceedings of the NASA Workshop on Atomic Oxygen Effects*, D. E. Brinza, Editor. JPL Publication 87-14, 1987. pp. 1-10.

Mende, S. B., Garriott, O.K., and Banks, P.M., Observations of Optical Emissions on STS-4, *Geophys. Res. Lett., 10*, 122, 1983.

Mende, S.B., and Swenson, G.R., Vehicle Glow Measurements on the Space Shuttle, in *NASA Conference Publication 2391: Second Workshop on Spacecraft Glow*, Edited by T. H. Waite, Jr. and T. W. Moorehead, Washington, DC. P.1, 1985.

Mende, S.B., G.R. Swenson, and E. J. Llewellyn, Space Vehicle Optical Contamination by Ram Glow, *Adv. Space Res., 7*, (5)169-(5)178, 1987.

Narcisi, R. S., Trzcinski, E., Federico, G., Wlodyka, L., and Delorey, D., The Gaseous and Plasma Environment Around Space Shuttle, AIAA 83-2659, Proc. AIAA, Washington, DC, October 1983.

Peters, P.N., Gregory, J.C., and Swann, J. ., Effects on Optical Systems from Interactions with Oxygen Atoms in Low Earth Orbit, *Appl. Optics, 25*, 1290, 1986.

Peters, P. N., R. C. Sisk, and J. C. Gregory, Velocity Distribution of Oxygen Atoms Incident on Spacecraft Surfaces, *J. Spacecraft Rockets, 25*, 53, Jan.-Feb. 1988.

Swenson, G. R., S. B. Mende, and K. S. Clifton, Ram Vehicle Glow Spectrum: Implication of NO_2 Recombination Continuum, *Geophys. Res. Lett., 12*, 97,1985.

Swenson, G. R., S. B. Mende, and K. S. Clifton, STS-9 Shuttle Glow: The Shuttle Ram Angle Effect and Absolute Intensities, *Geophys. Res. Lett., 13*, 509-512, 1986.

Wulf, E., and von Zahn, U., The Shuttle Environment: Effects of Thruster Firings on Gas Density and Composition in the Payload Bay, *J. Geophys. Res., 91*, 3270, 1986.

CONTAMINATION OF SPACECRAFT BY RECONTACT OF DUMPED LIQUIDS

by

M. E. Fowler*, L. J. Leger*,
M. E. Donahoo**, and P. D. Maley***

Liquids partially freeze when dumped from spacecraft producing particles which are released into free space at various velocities. Recontact of these particles with the spacecraft is possible for specific particle sizes and velocities and, therefore, can become contamination for experiments within the spacecraft or released experiments as a result of waste and potable water dumped from Space Shuttle. An examination of dump characteristics was conducted on STS-29 using both on-board video records and ground based measurements. A preliminary analysis of data from this flight indicates particle velocities are in the range of 30 to 75 ft/sec and recontact is possible for limited particle sizes.

* Materials Branch, Johnson Space Center
** Orbit Analysis Branch, Johnson Space Center
*** Rockwell International

INTRODUCTION

Several times during a mission, the Space Shuttle Orbiter has to release water which has accumulated from the fuel cells. Due to the vacuum environment into which they are being released, the water flash evaporates, leaving small ice particles. There has been some concern that the motion imparted to these particles could return them to the orbiter during subsequent orbits. This could lead to contamination of experiments in the payload bay and in the extreme, damage to materials such as the thermal protection system tiles. It has been reported that particles contacted the orbiter on STS-8 and STS-61A during subsequent orbits after water dumps (1). As a consequence, water dumps are being planned more carefully with respect to the orientation of the orbiter during the dump. Judicious angles at which the water is released can lead to the water reentering (7) the earth's atmosphere prior to recontact with the vehicle.

A Detailed Test Objective (DTO) was formulated for STS-29 where the water dumps were planned around the viewing opportunities from both the orbiter and ground cameras. The idea was to release the water as the orbiter was first coming into light over the event horizon yet still against a black background for better viewing. Such observations would provide general plume shape and therefore velocity vector information for the released particles. The video images from the orbiter provided the means to analyze the particles' trajectories and velocities in the near field. With this information, it was possible to analyze the orbit of the particles and determine if recontact was possible. With this information, a model could be developed for use in detailed mission planning.

ICE PARTICLE ANALYSIS

The first step to the analysis of the recontact problem involved understanding the ice particle flow including velocity distribution as well as the mass of the particles. The stream of water released from the orbiter is very collimated up to the burst zone (2,3). The burst zone is mostly regulated by the vapor pressure of the liquid, in this case water, as the liquid begins to boil (3). Adding dissolved gasses can decrease the length of the collimated stream (4-6) due to the expansion of the gases in vacuum. Other factors involved include the orifice diameter, temperature of the liquid, and initial pressure of the stream before release (2-6). In this case, the burst zone is approximately 1-2 ft away from the orbiter.

The size of the ice particles formed can be estimated (4) by using a force balance on the water droplet before evaporation occurs

$$D = 30\gamma/P_v \qquad (1)$$

where D is the diameter in microns, γ is the surface tension of the liquid in dynes/cm, and P_v is the vapor pressure. Table 1 gives typical diameters for water

Table 1 Ice particle size vs temperature

Temperature °C	D (μ)
0	479
10	238
25	92.3
30	68.9
50	23.7

at various temperatures. These values are only estimates as they use the surface tension of pure water in air at 18°C. A typical potable water dump is made at temperatures between 25 and 30°C and approximately 15-20 psia at the nozzle, and consists of pure water. Waste water releases also occur under the same conditions..

The velocity of the particles was determined from the video tape taken on board of the dump which occurred on orbit 49 (March 16, 1989). While time consuming, the method was relatively straight forward. A particle was first found that could be differentiated from the flow through several fields of the video image (there are 2 fields per frame). If a part of the orbiter structure was in view, the dimensions of the structure were used as a reference in order to find the distance traveled by the particle. Simple geometry is used to estimate the flight of the particle over a given number of fields (one field = 1/60 sec.). In other cases where the orbiter was not in view, the downlinked data on the movement of the camera about a pivot allowed the same calculations to be made.

From these calculations, the relative velocity of the particles were found to be between 30 and 75 ft/sec. The angle at which the particles left the orbiter were not completely definable with the video images available, but for this analysis will be assumed to be perpendicular to the orbiter out the port side, parallel to the y-axis of the orbiter. The actual plume is roughly conical in form. It should also be noted that the actual size of the particles could not be determined with any accuracy in the video images but appear to be larger than the theoretical. This might be possible if the structure of the ice particles were more porous like snowflakes than solid ice spheres. Trapped gasses and vapor expanding in the particles could account for larger than theoretical sizes.

ORBITAL ANALYSIS

All of the orbital mechanics analysis of the ice particles was performed using software developed at the Johnson Space Center called High Accuracy Relative Motion Processor (HAREM). The HAREM program takes the atmospheric model for the date that is specified, the shuttle weight, altitude and orbit, along with the particle parameters such as relative velocity vector to the orbiter, weight, center of mass, and moments of inertia, and predicts the orbit of the particles with respect to the orbiter. A parametric study was made of such parameters as ice mass and velocity vector in order to determine if recontact was possible for the water dump of STS-29.

Table 2 Particle size and weight

Diameter (cm)	Weight (mg)
0.01	5.24×10^{-4}
0.1	0.524
1.0	524

Table 2 describes the particles of interest for this study. The particle sizes were chosen to encompass the theoretical particle sizes plus a larger particle which might be formed due to vapor expansion in accordance with the video tape images. The relative speeds used for this study were 12, 30 and 75 ft/sec. The orbiter was positioned in such a way as to release the particles in several directions; retrograde, posigrade, 90° pure out of plane, 90° radially up and down, and 45° out of plane relative to a local vertical, local horizontal (LVLH) reference plane. The orbiter was considered to be in an orbit 160 x 160 nautical miles. The date was taken to be March 1, 1989, or roughly the date of STS-29, unless otherwise noted.

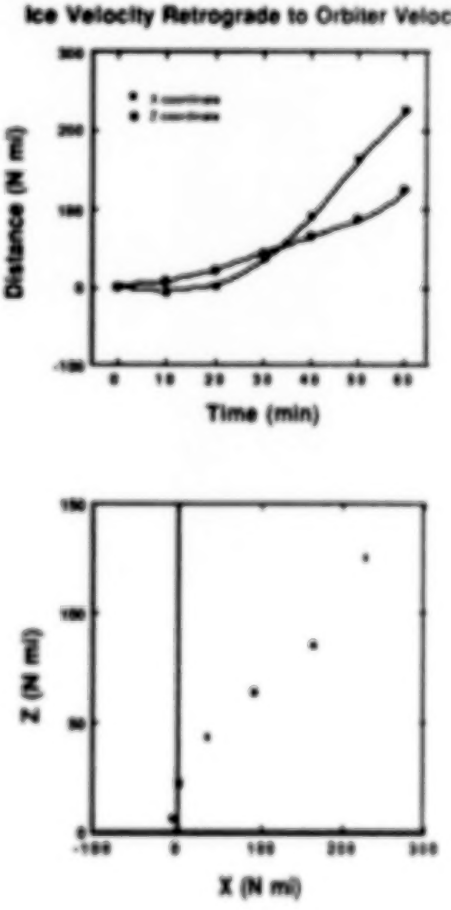


Figure 1 One mm particle in retrograde motion release at 30 ft/sec

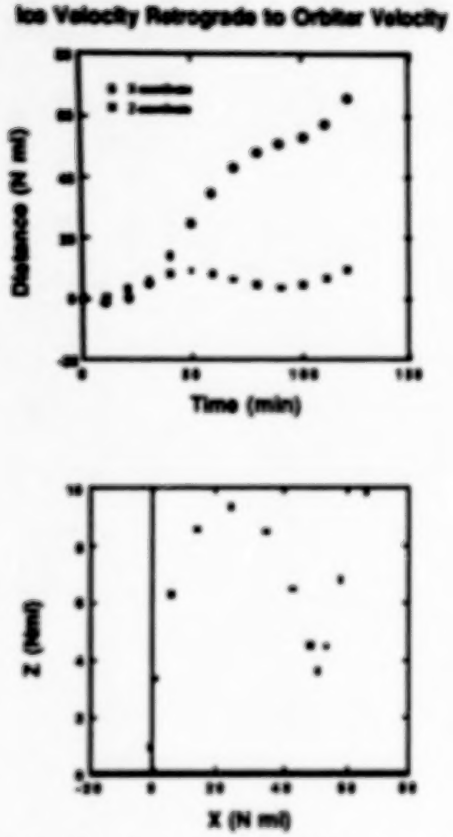


Figure 2 One cm particle in retrograde motion released at 75 ft/sec

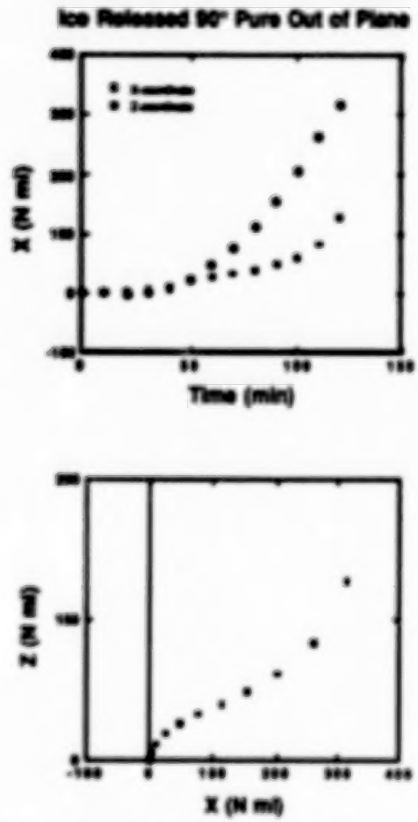


Figure 3 One mm particle released 90° pure out of plane at 75 ft/sec

In Figures 1 and 2, two different graphs are used to show the relative motion of the ice particles with respect to the orbiter. The first figure plots the X- and Z- coordinates with respect to the orbiter centered LVLH coordinate frame as a function of time. The second plots the Z- coordinate as a function of the X- coordinate in the LVLH reference frame. In the LVLH reference frame, +Z is down toward the earth, and the origin of this plot is therefore the orbiter. As can be seen in these two figures, a retrograde release of the ice particles, regardless of the size of the particle, results in deorbit of the particles in a timely fashion, posing absolutely no threat to the orbiter. This should not be surprising since a retrograde velocity slows the particles, causing them to fall to a lower orbit, where drag forces cause further decay of the orbit. It should be noted that the larger particle stays in orbit longer than the smaller particle, and can be explained in terms of the energy of the orbit. Given the same relative velocity, the energy of the particle is determined by its mass, thus the larger particle has more energy and stays in orbit longer. This poses interesting possibilities for decreasing the orbit time of the particles by decreasing their size.

Figures 3-5 represent the orbits of particles of 1 mm in diameter which are 90° pure out of plane, 90° radially up and 90° radially down to the orbiter in LVLH. As can be seen in these figures, recontact does not seem to be possible before reentry. As a check, a 1 cm particle was released at 90° pure out of plane at 30 ft/sec and 75 ft/sec and are depicted in figures 6 and 7 respectively. In both cases, the particles are well on their way to reentry within two orbits. This is not meant to imply that no conditions for these particular release angles will ever cause recontact, but this limited evidence is promising.

Trajectories for a 1 mm particle released in a posigrade trajectory at 30 ft/sec are shown in figure 8. This particle, due to its small diameter does not recontact the orbiter. Its orbit decays below that of the orbiter after two orbits. Figure 9 represents the case of a 1 cm particle traveling at 75 ft/sec when it leaves the orbiter. As can be immediately seen, this particle stays in orbit much longer than the previous cases, traveling in an orbit above the orbiter initially. While recontact is not observed, it shows that

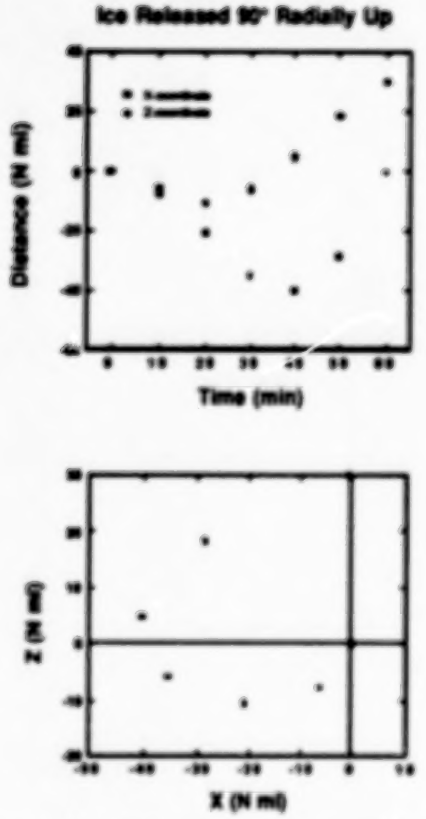


Figure 4 One mm particle released at 30 ft/sec away from the earth

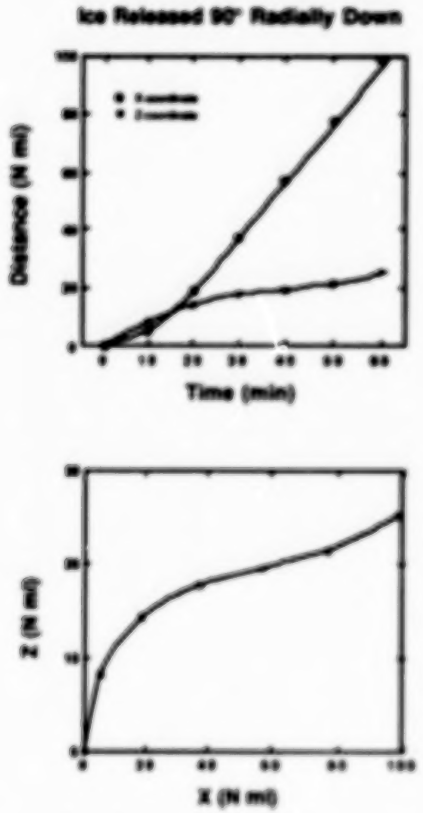


Figure 5 One mm particle released at 75 ft/sec toward the earth

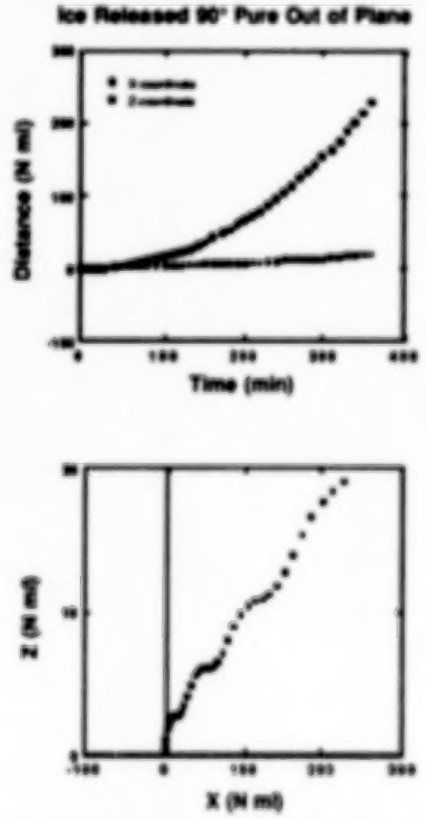


Figure 6 One cm particle released 90° pure out of plane at 30 ft/sec

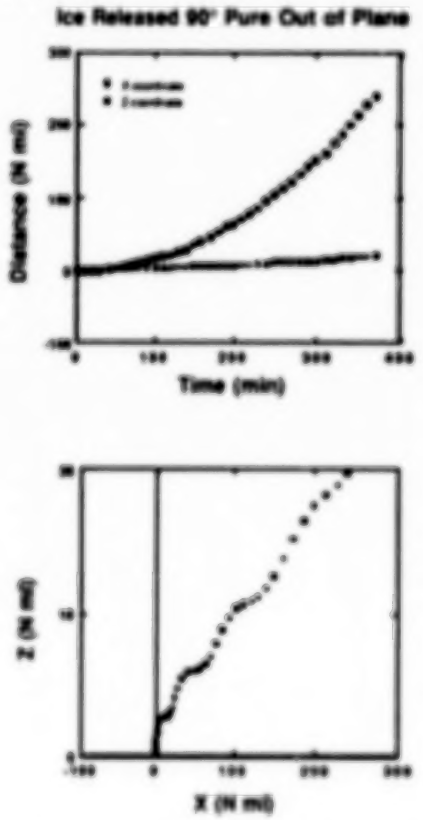


Figure 7 One cm particle released 90° pure out of plane at 75 ft/sec

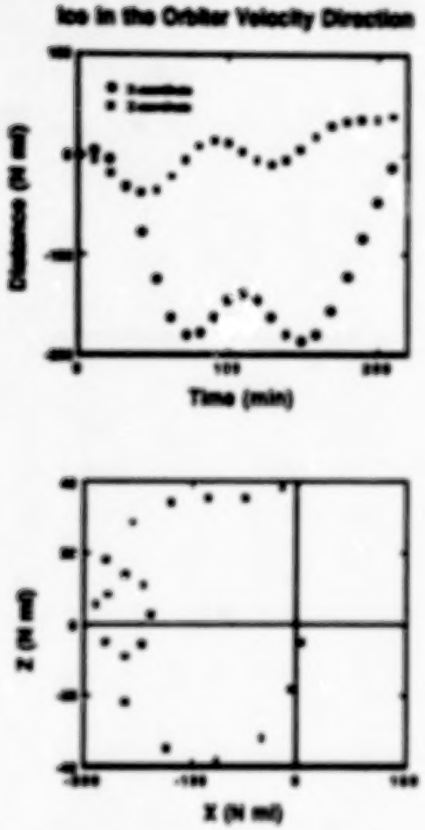


Figure 8 One mm particle in posigrade motion at 30 ft/sec

there are orbits which could cause problems. Given that recontact was observed for STS-61A, the same particle and trajectory was analyzed for the November time frame of 1985, or the approximate time of STS-61A. The atmosphere should have been less dense at this altitude for this time frame since the solar cycle is just now reaching its peak in the 1989-1990 time frame. The trajectory results are found in Figure 10. At 75 ft/sec, a 1 cm particle stays in orbit, generally at an X-coordinate many miles from the orbiter. The ice particles, however, oscillate in the Z-coordinate around the orbiter's orbit. Somewhere around the tenth orbit after release, recontact becomes possible as the ice particles begin to catch up to the orbiter. Apparently, the higher density of the atmosphere of 1989 creates enough drag to deorbit the particles faster.

With this in mind, it follows that other combinations of speed and mass would produce recontact given a posigrade trajectory. Figure 11 illustrates recontact for the case of a 1 cm particle with an initial velocity of 12 ft/sec in a posigrade motion with the 1985 atmospheric model. In this case, recontact is possible much sooner, with opportunities twice in the first 2 orbits. Slight angles to the pure posigrade trajectory may also facilitate recontact.

While this study covers only limited particle size and velocity distributions, it points out the possibilities for recontact given the right conditions. Retrograde trajectories appear to preclude particle recontact.

CONCLUSIONS

It was found that a larger particle has a longer time in orbit. Breaking up the particles into smaller spheres will have some effect on the decay time and therefore provides another means by which recontact can be avoided.

The atmospheric density also plays a key role in the decay of the particle orbits. It was shown that the 1985 atmosphere had less of an effect on the orbit of the particles than the 1989 atmosphere. Since density of the atmosphere is a changing function of time, it is another parameter to take into account for the entire problem.

Release of water in posigrade trajectories was found to recontact the orbiter and could cause problems for experiments

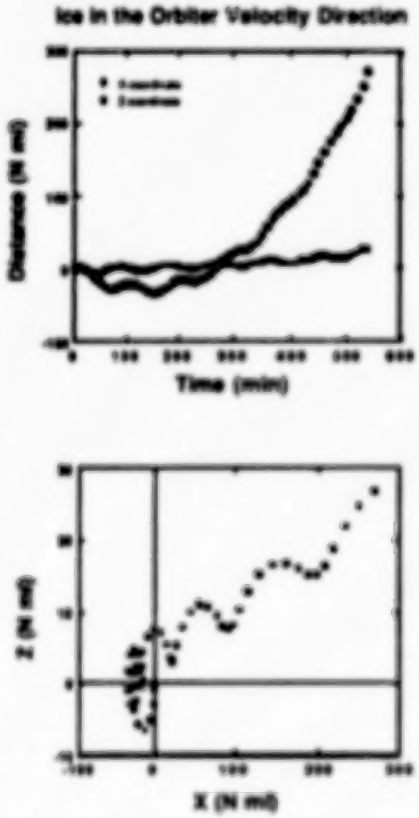


Figure 9 One cm particle released at 75 ft/sec in a posigrade motion

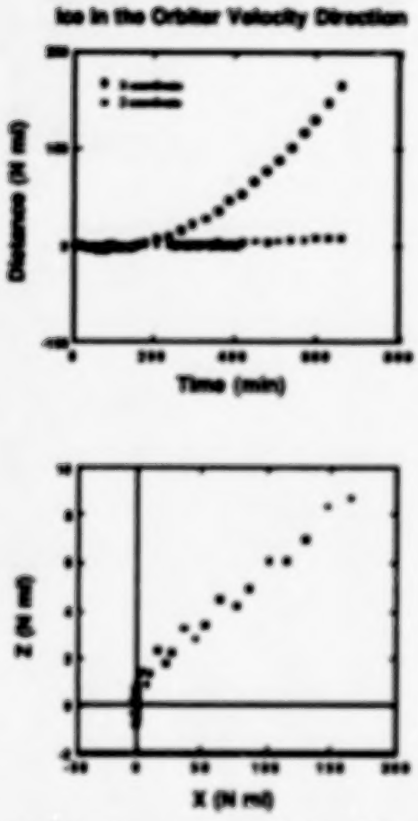


Figure 11 One cm particle (12 ft/sec) in a posigrade motion for the 1985 atmosphere

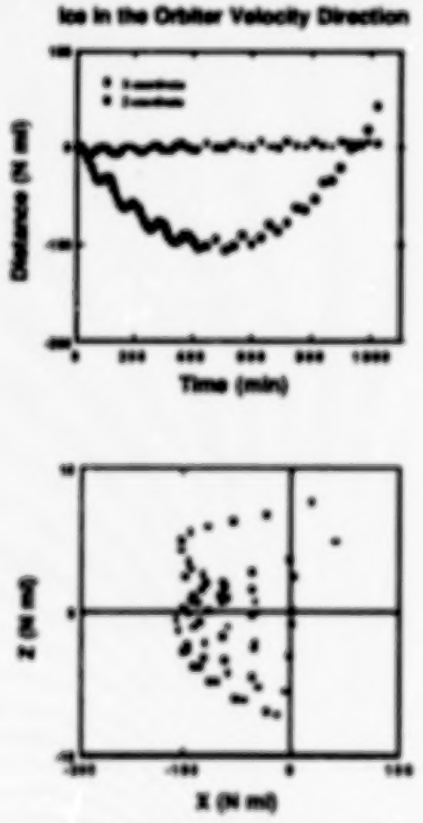


Figure 10 One cm particle released in a posigrade motion at 75 ft/sec (1985)

operating in attached payload mode or even for certain cases of release payloads.

Additional study of particle characteristics is needed to preclude deposition of material (residue from impact) on experiment surfaces and field-of-view interference.

ACKNOWLEDGEMENTS

The authors wish to thank Gene Beck of G.E. for his efforts with the video imaging. Many thanks also go to Joel Montalbano of Rockwell International for his tutelage with HARDEM.

REFERENCES

1. Communication with Mission Specialist Jim Buchli.

2. H. Fuchs and H. Legge, Acta Astronautica, 6, 1979, 1213.

3. E.P. Muntz and M. Orme, AIAA Journal, 25, 1987, 746.

4. J. Gayle, C. Egger, and J. Bransford, J. Spacecraft, 1, 1964, 323.

5. R. Mikatarian and R. Anderson, J. Spacecraft, 3, 1966, 267.

6. D. Bharathan and T. Penney, J. Heat Transfer, 106, 1984, 407.

7. R. Naumann, "Dynamics and Column Densities of Small Particles Ejected From Spacecraft," NASA TN D-7590, Feb. 1974.

SPACECRAFT GLOW CURRENT STATUS

S. Mende and G. Swanson
Lockheed Missiles and Space Company

(Paper not provided by publication date.)

AUTOMATION IN THE SPACE STATION MODULE POWER MANAGEMENT AND DISTRIBUTION BREADBOARD

Bryan Walls and Louis F. Lollar
Marshall Space Flight Center, EB12
Huntsville, AL 35812

ABSTRACT

The Space Station Module Power Management and Distribution (SSM/PMAD) Breadboard, located at NASA's Marshall Space Flight Center (MSFC) in Huntsville, Alabama, models the power distribution within a Space Station Freedom Habitation or Laboratory module. Originally designed for 20 kHz ac power, the system is now being converted to high voltage dc power with power levels on a par with those expected for a space station module [1].*

In addition to the power distribution hardware, the system includes computer control through a hierarchy of processes. The lowest level process consists of fast, simple (from a computing standpoint) switchgear, capable of quickly safing the system. The next level consists of local load center processors called Lowest Level Processors (LLP's). These LLP's execute load scheduling, perform redundant switching, and shed loads which use more than scheduled power. The level above the LLP's contains a Communication and Algorithmic Controller (CAC) which coordinates communications with the highest level. Finally, at this highest level, three cooperating Artificial Intelligence (AI) systems manage load prioritization, load scheduling, load shedding, and fault recovery and management. The system provides an excellent venue for developing and examining advanced automation techniques. This paper examines the current system and the plans for its future.

INTRODUCTION

As the electrical power requirements for spacecraft have increased, the problems of managing these large systems have also increased. Since 1978, NASA/MSFC has been actively working the problem of spacecraft power system automation. This work has progressed from reference power system studies to operating test beds employing both conventional and expert system computer controls. One of these systems is the SSM/PMAD Breadboard [2].

*Reference 1 is used throughout this paper unless otherwise noted.

SYSTEM DESCRIPTION

The automation studies which lead to the SSM/PMAD Breadboard began at MSFC in 1984. A primary purpose of the breadboard is to investigate automation techniques appropriate to a large PMAD system such as will exist on Space Station Freedom. The current SSM/PMAD Breadboard consists of the 20 kHz power distribution hardware, the automation and control software, and the computer hardware shown in Figure 1.

Power Distribution Hardware

A typical configuration of the breadboard is shown in Figure 1. The 20-kHz, 208-Vac, single-phase power is supplied to both of the Power Ring Buses by a 3-kW Mapham-type power supply. The distribution system contains three types of switches: the Remote Bus Isolators (RBI's), the Remote Controlled Circuit Breakers (RCCB's), and the Remote Power Controllers (RPC's). The loads include light bulbs (two 150 W bulbs in series for each of four loads), 12 resistive loads adjustable to 1250 W in 250-W increments, and several low-power LED's. Further, the components of a given type are interchangeable to allow testing of different system configurations. The shaded areas in Figure 1 denote sections of the system for which no hardware is yet available, but the following system components descriptions will assume full system capabilities.

The Ring Bus architecture allows hardware to be powered despite any failure of a single RBI, and permits a section or sections of components to be isolated from a powered bus by the RBI's. Each Ring Bus contains three 15-kW RBI's, shown as black diamonds in Figure 1. To avoid problems with current sharing and power flow, only two RBI's on a given Ring Bus may be closed at any given time. The RBI's are not designed to be opened while current is flowing through them.

The RCCB's are shown as square white boxes in Figure 1. Each RCCB contains a remotely controlled mechanical switch, rated at 10 kW, which will open automatically on an I^2t condition. In addition, each RCCB contains current-sensing electronics which can report current levels, switch status, and control status on request.

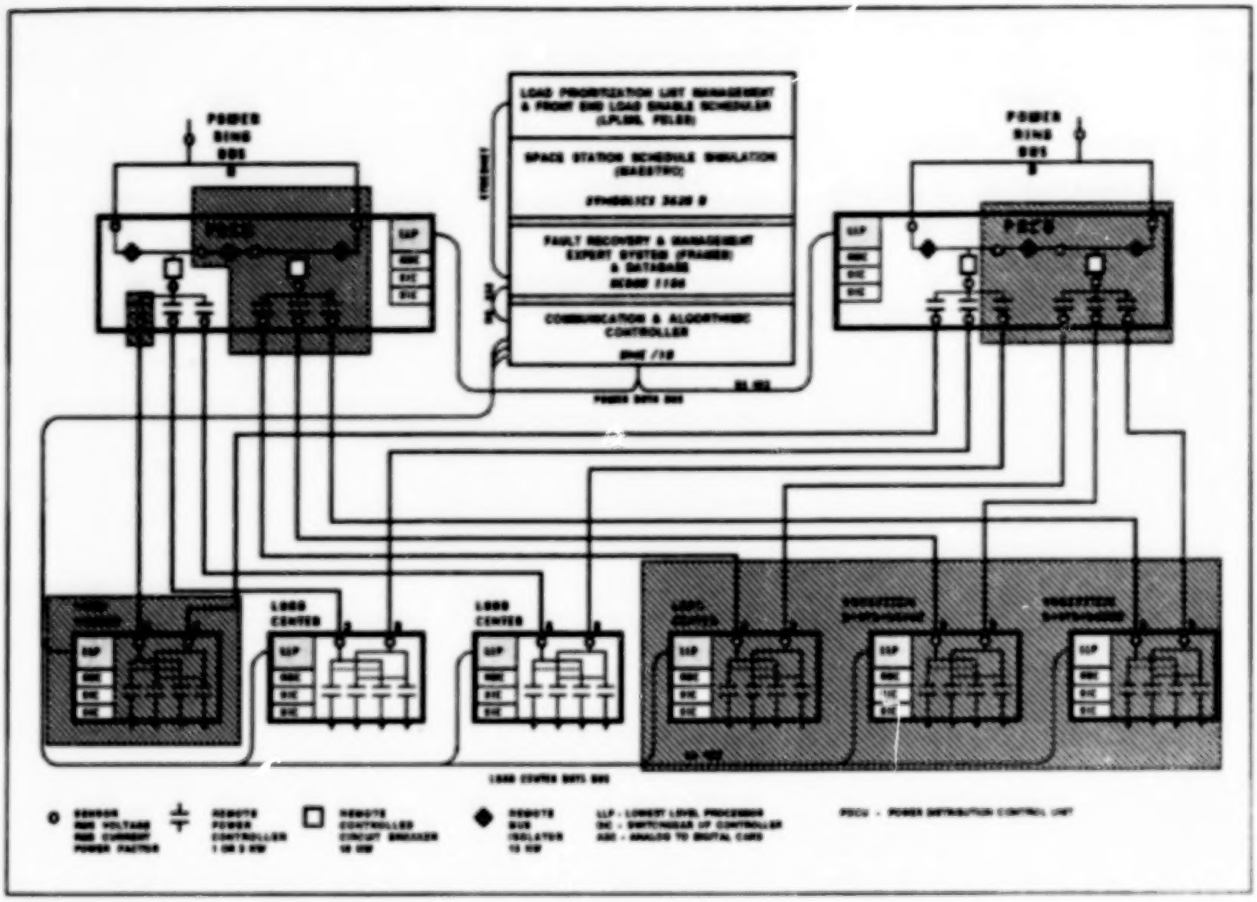


Figure 1 - Current SSM/PMAD Configuration (20 kHz Ring Bus)

Both of the Power Distribution Control Units (PDCU's) contain six 3-kW RPC's, three below each of the RCCB's. The 3-kW RPC's are also used in the Subsystem Distributors while 1-kW RPC's are used in the Load Centers. The 1-kW and 3-kW RPC's differ only in their current ratings and trip levels. The RPC's are similar to, but somewhat more sophisticated than, the RCCB's. In addition to the RCCB capabilities, RPC's also provide current limiting and can trip on under voltages, immediate overcurrents, and ground faults. The same relay symbol is used for both 1-kW and 3-kW RPC's in Figure 1.

Finally, the planned locations of all sensors are represented as white circles in Figure 1. These sensor packages exist throughout the system with each package containing a voltage and current sensor allowing for RMS voltage and current, average power, and power factor calculations. Due to the current values available from the RCCB's and RPC's, the automation software now makes limited use of the sensor readings; thus only a few of the sensors are installed.

Software and Platforms

In addition to the power distribution hardware, the system includes computer control through a hierarchy of processes. Each step up the hierarchy shows a decrease in speed (from microseconds at the lowest level, to milliseconds or seconds at the middle level, to seconds or minutes at the highest level) and an increase in sophistication. The lowest level process consists of fast, simple (from a computing standpoint) switchgear, capable of quickly safing the system. The next level consists of local load center processors called Lowest Level Processors (LLP's). These LLP's execute load scheduling, perform redundant switching, and shed loads which use more than scheduled power. The level above the LLP's contains a CAC which coordinates communications with the highest level. Finally, at the highest level, three cooperating AI systems manage load prioritization, load scheduling, load shedding, and fault recovery and management.

The LLP's are at the level nearest the power hardware and consist of Motorola MVME 107 single-board 68010 based computers, each with an RS422 communications board. Each LLP communicates over RS422 to the power hardware through one or two Switch Interface Cards (SIC's), which in turn communicate with the RPC's and the Analog to Digital Converter Cards for sensor data packets. Each lowest level domain — Load Center, Subsystem Distributor, and PDCU — contains one LLP. Each LLP is responsible for controlling switches and monitoring all of the sensor readings and switch positions in its lowest level domain. The LLP also executes scheduled changes in switch positions, sheds any loads which exceed their scheduled maximum, and switches redundant loads to their secondary bus (in a Load

Center or Subsystem Distributor) if the load's primary source is interrupted.

The LLP notifies the next higher machine in the hierarchy, the CAC, of any anomalies noted. The CAC routes information to the various LLP's, provides the source code which is downloaded to the LLP's when the system is initialized, and serves as the control station when the breadboard is operated in manual mode. Messages pass between the CAC, the LLP's, and either the Fault Recovery and Management Expert System (FRAMES), Maestro, or the Load Priority List Maintenance System (LPLMS). The CAC is resident on a Motorola VME-10 computer and communication is over RS422 to the LLP's and over RS232C to a Xerox 1186 for the others.

FRAMES, Maestro, and the LPLMS share the highest level of the software hierarchy. FRAMES monitors the system for anomalies. Maestro is a resource scheduler which can create a schedule based on multiple constraints. The LPLMS keeps up with the dynamic priorities of all payloads and develops load shedding lists for contingencies which require load shedding. Each of these three systems is described below.

FRAMES is responsible for detecting faults, advising the operator of appropriate corrective actions, and in many cases autonomously implementing corrective actions through power system reconfiguration [2]. FRAMES and the LLP's concurrently receive a schedule from Maestro. Then, FRAMES receives notification of any anomalies, such as tripped breakers or shed loads, from the LLP's. Messages containing sensor readings are also sent to FRAMES. Next, FRAMES uses this information and attempts to find an explanation for any anomalies. If this explanation requires removing some pieces of equipment from service, FRAMES does so and notifies Maestro to adjust the schedule accordingly. Finally, FRAMES shows schematically the results of the anomaly, explains to the user the reasoning behind these results, and waits for notification of further anomalies.

Maestro is a multiple constraint resource scheduler. The constraints currently being used in the SSM/PMAD Breadboard include crew member requirements, equipment resources, and power resources. In this breadboard, power is the resource of most concern. Power is allocated by the amount available to the whole system and by the ability of intervening components to supply the power, e.g., multiple 1-kW RPC's below a single 3-kW RPC [2].

A user selects a number of activities from the activity library and requests that they be scheduled. Then Maestro creates an initial schedule for the system. An activity is made up of a task name, a base priority of the task, the number of times the task should be repeated, and a collection of one or more subtasks. The powered equipment is chosen from the equipment library. The powered equipment description includes how much power it is allowed, whether it may be tested by the system (have power toggled on and off), where it may be connected, and whether it can be redundant. Elements may be added to the Activity or Equipment Libraries by using the appropriate editor. The activities are scheduled, according to their priority, such that no constraints are violated. From the schedule, 30-minute sections of the complete schedule, called event lists, are created. An event list shows when each switch should be turned on or off, how much current it is allowed to pass, whether it it testable, and whether it can switch to redundant. A new event list is created every 30 minutes, unless some anomaly causes a contingency list to be created within that time. A contingency list includes a new event list and a new Load Priority List which resets the timer.

The third of the AI systems is the LPLMS. The LPLMS uses information from the event list and the activity library, along with its own rules, to dynamically assign relative priority to each active load in the system. The load priority list is used to shed loads in case of power reductions. A new list is sent to the LLP's at least every 15 minutes (less than 15 if a contingency occurs).

SYSTEM MODIFICATIONS

Some major changes are planned for the SSM/PMAD Breadboard. Work is now under way to change the system from the current 20 kHz 208 Vac Ring Bus configuration to a 150 Vdc Star Bus configuration. In the automation and control area, there will be a change in the hardware platforms for the LLP's, CAC, and FRAMES, and an upgrade to the communications so that Ethernet can be used throughout. Figure 2 illustrates these modifications. A new Knowledge Based Management System (KBMS) will be introduced into the system, as well as a centralized enhanced model. An intermediate level of autonomy will be added so that "expert help" will be available to the operator. Finally, SSM/PMAD will be connected to Lewis Research Center's Autonomous Power System (LeRC APS).

Power Hardware Changes

The change to a 150 Vdc Star Bus topology on the breadboard followed modifications in the Space Station Freedom baseline. As Figure 2 shows, the change is most pronounced in the simplified PDCU's. The change to dc requires replacing all of the switches, except the RBI's, and much of the wiring. Although switching the dc current is more difficult, the logic associated with each switch can be reduced, since there is no need to detect zero crossings or phase angle. In addition, Subsystem Distributors are no longer required and sensors packets will only consist of current and voltage sensors.

The conversion to the Star Bus Topology will remove the requirement for RCCB's. Initially the current RBI's will be used, until a new 25-kW Remote Bus Isolator, capable of switching current, is developed and added to the system. The topology has much more impact on the software than does the dc change. Maestro, FRAMES, and the CAC all use power system topology information in their operations and it is this distribution of similar information which indicates the need for a centralized, enhanced model of the system. Until that model is completed, the code will be modified to keep the current capabilities.

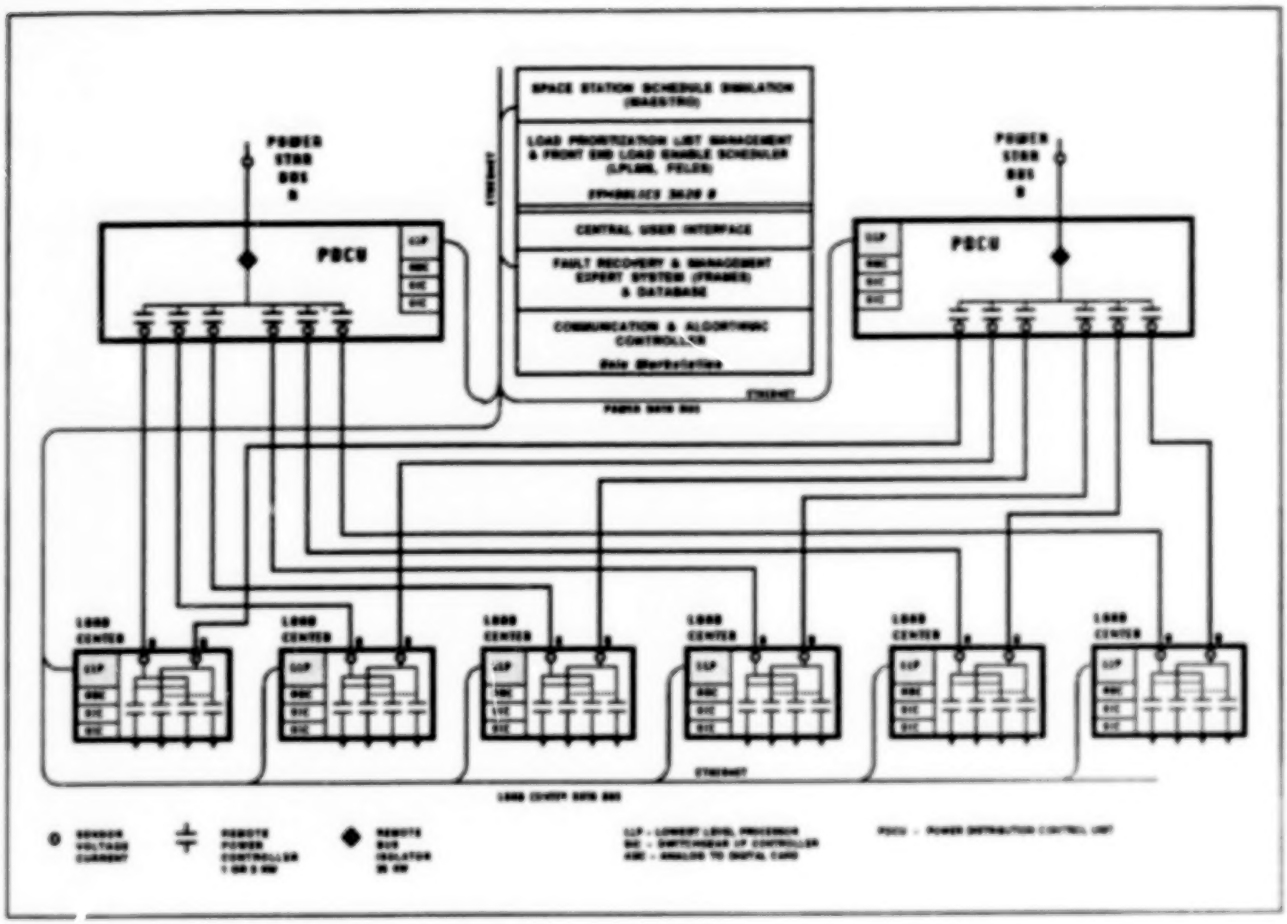


Figure 2 - Planned SSM/PMAD Configuration (dc Star Bus)

Automation Hardware Upgrades

Some computer and communications upgrades are in process for the breadboard. The RS232C link between the platforms for FRAMES and the CAC has been a bottleneck. The CAC's inability to communicate with more than one LLP at a time has been an even worse problem. Also, while running FRAMES, the Xerox 1186 Workstation is operating at its limit. It cannot handle the planned improvements and additions to FRAMES. For these reasons, it was decided to replace both the Xerox 1186 and the CAC's platform, a Motorola VME-10, with a single, high power workstation. Today's workstations are very capable in both computing and communicating. A machine comparable to a Sun 4 should be able to host the current CAC and FRAMES functionality with significant resources left over for development of the KBMS and the Enhanced Model. Since the software for the CAC is written in Pascal, and FRAMES is written in Common Lisp using the Common Lisp Object System (both available on most common UNIX based workstations), software porting should be relatively straightforward. An added advantage of the new Workstation is the availability of relatively inexpensive color graphics. A central user interface would be a big plus when operating in manual or semi-manual mode, as well as for monitoring fully autonomous operation.

Combining the CAC and FRAMES on a single platform will remove the RS232C bottleneck while communications to the Symbolics will continue to be via Ethernet. Ideally, communications to the LLP's should also be by Ethernet. The 68010 processors in the current LLP's are capable of supporting their current utility and the addition of Ethernet communications. However, the cost of adding Ethernet boards to the existing VME-bus backplane is higher than getting new 80386 based computers with new rack-mount cabinets, Ethernet boards, floppy disk drives, monitors, and keyboards. Because of lower cost, the vast amount of software available for 80386 machines, and the Space Station Freedom baseline of the 80386 for onboard processing, the decision was made to purchase new 80386 based computers for the LLP's.

Other Changes

The KBMS is now under development. The rules in each of the AI systems will be organized into modular groups with the KBMS controlling rule execution and managing modification of the rule bases. In addition, the KBMS will control operation of the user interface.

Working with the KBMS will be the Enhanced Model. This causal model will support a better user interface, provide for more general diagnostic capabilities, serve as a basis for simulated fault

injection and "what if" capabilities, allow for construction of various topologies fairly easily, permit a more natural representation of constraints in the domain of power systems, and enable natural growth of power system fault diagnosis and management. In addition, it could allow easy domain adjustments and upgrading, allow new techniques in reasoning to be used, and would make domain knowledge as it exists in the knowledge base easier to develop and manage.

In the current system, control is either fully autonomous, once the desired activities are chosen and the breadboard started, or fully manual through a rudimentary monochromatic menu system. For the modified system, layers of intermediate autonomy will be developed so that the information contained in the system will be available at the level desired by the user.

Since the SSM/PMAD Breadboard is required to support the development of the Power Management and Distribution system for the space station modules, Boeing Aerospace Company, the prime contractor for Work Package #1 at MSFC, will be the primary user of the system. Therefore, continuing breadboard advanced development will be on a noninterference basis with the Boeing work. Finally, since Lewis Research Center (LeRC) is responsible for the space station power generation, storage, and primary distribution, the SSM/PMAD Breadboard will be interfaced with the LeRC Autonomous Power System Demonstration Program to help ensure that the two will be well integrated on Space Station Freedom.

CONCLUSIONS

This paper has presented an overview of the current status of the Space Station Module Power Management and Distribution Breadboard, and a glance at the plans for the future. Testing has demonstrated that, though certainly still a development system, the breadboard is quite mature in its ability to operate autonomously and to correctly react to many power system faults.

The current breadboard is about to undergo major revisions which will permit the system to be even more capable and mature. The change to a dc Star topology brings the breadboard more in line with plans for Space Station Freedom modules. Development of cooperating expert system technology, KBMS, and Enhanced Modeling will keep the breadboard on the leading edge of spacecraft power automation. Continued use by MSFC and Boeing and joint projects with LeRC will all contribute to make the SSM/PMAD Breadboard a valuable resource to MSFC, NASA, and the world.

REFERENCES

1. Walls, Bryan, "Exercise of the SSM/PMAD Breadboard," Proceedings of the 24th IECEC, publication pending, August 1989.

2. Lollar, Louis F. and Weeks, David J., "The Autonomously Managed Power Systems Laboratory," Proceedings of the 23rd IECEC, Vol. 3, August 1988, pp. 415-419.

FURTHER READING

Ashworth, Barry, "An Architecture for Automated Fault Diagnosis," Proceedings of the 24th IECEC, August 1989.

Britt, Daniel L., Gohring, John R., and Geoffroy, Amy L., "The Impact of the Utility Power System Concept on Spacecraft Activity Scheduling," Proceedings of the 23rd IECEC, Vol. 3, pp. 621-626, August 1988.

Freeman, Kenneth A., Walsh, Rick, and Weeks, David J., "Concurrent Development of Fault Management Hardware and Software in the SSM/PMAD," Proceedings of the 23rd IECEC, Vol. 3, pp. 307-312, August 1988.

Miller, William D., et al., "Appendix X: SSM/PMAD Expository and Activity Plan," Space Station Automation of Common Module Power Managment and Distribution Interim Final Report, Martin Marietta Aerospace Denver Astronautics Group, Denver, CO, pp. X1-X31, February 1989.

Miller, William D., and Jones, Ellen F., "Automated Power Management Within a Space Station Module," Proceedings of the 23rd IECEC, Vol. 3, pp. 395-399, August 1988.

NASA SYSTEMS AUTONOMY DEMONSTRATION PROJECT: ADVANCED AUTOMATION DEMONSTRATION OF SPACE STATION FREEDOM THERMAL CONTROL SYSTEM

Jeffrey Dominick
Crew and Thermal Systems Division
NASA - Johnson Space Center
Houston, Texas

John Bull
Information Sciences Division
NASA - AMES Research Center
Moffett Field, California

Kathleen Healey
Systems Development
and Simulation Division
NASA - Johnson Space Center
Houston, Texas

Abstract

Congress has displayed substantial interest in accelerating the dissemination of advanced automation technology to and in U.S. industry. Space station was selected as the high-technology program to serve as a highly visible demonstration of advanced automation, and spur dissemination of the technology to the private sector.

The NASA Systems Autonomy Demonstration Project (SADP) was initiated in response to the above stated Congressional interest for Space station automation technology demonstration. The SADP is a joint cooperative effort between Ames Research Center (ARC) and Johnson Space Center (JSC) to demonstrate advanced automation technology feasibility using the Space Station Freedom Thermal Control System (TCS) test bed.

A model-based expert system and its operator interface have been developed by knowledge engineers, AI researchers, and human factors researchers at ARC working with the domain experts and system integration engineers at JSC. Its target application is a prototype heat acquisition and transport subsystem of a space station TCS.

The demonstration is scheduled to be conducted at JSC in August, 1989. The demonstration will consist of a detailed test of the ability of the Thermal Expert System to conduct real time normal operations (start-up, set point changes, shut-down) and to conduct fault detection, isolation, and recovery (FDIR) on the test article. The FDIR will be conducted by injecting ten component level failures that will manifest themselves as seven different system level faults.

This paper describes the SADP goals, objectives, and approach; it describes the Thermal Control Expert System that has been developed for demonstration, and provides insight into the lessons learned during the development process.

Introduction

The NASA Systems Autonomy Demonstration Program (SADP) was initiated in response to Congressional interest for space station automation technology demonstration.[1] The technical objectives of SADP are to:

- Develop and validate knowledge-based system concepts and tools for real time control of a complex physical system
- Demonstrate enhancements to a space system's performance through advanced automation.

The programmatic objectives of SADP are to:

- Establish in-house expertise and facilities.
- Transfer advanced automation technology to operational centers.

Managed out of the Ames Research Center (ARC), SADP began its first joint cooperative project in 1986 with the Johnson Space Center (JSC), in an effort to transfer the expert system technology under development at ARC to a space station operations center. Ames is providing expertise in knowledge engineering, operator interfaces, and system architectures. Johnson Space Center is providing expertise in systems integration and in thermal engineering domain expertise.

The Space Station Thermal Control System (TCS) test bed at JSC was selected as the project application focus because it had several test articles under development, each requiring real time control and fault detection that an expert system could potentially provide. This paper gives an overview of SADP's Thermal Expert System (TEXSYS) project and describes some lessons learned.

Technology Challenge

The key technology challenge for TEXSYS is to provide real time control of a large electro-mechanical system. The TCS Heat Acquisition and Transport Subsystem is a complex physical system utilizing advanced thermal technology.

The specific prototype test article for the TEXSYS demonstration uses two-phase anhydrous ammonia as the coolant fluid, and consists of 5 evaporators, 4 condensers, 2 accumulators, a pump, 17 isolation valves, and numerous pressure-temperature sensors.

Comparison To Conventional Systems

Conventional control systems used by thermal engineers provide monitoring of system parameters, automatic control of nominal operations (startup, temperature setpoint changes, and shutdown), and notification of the operator when a parameter exceeds predetermined limits. The operator then has the task of analyzing the system situation to determine the best course of action. On the other hand, TEXSYS provides automatic control of nominal operations, monitors the system performance, and in addition, has the knowledge to analyze the data, take action to recover, and explain to the operator the fault diagnosis and reasons for actions taken. By elevating the task of the thermal engineer to a higher level of system monitoring and tasking, it is anticipated that operator performance and productivity will be enhanced.

Expert System Technology Thrusts

In response to the TCS challenge, the project's expert system technology development has been concentrated in the following areas: (1) integration of knowledge-based systems into a complex real time environment; [2] (2) causal modeling of complex components and elements through representation of first principles, quantitative models, and qualitative models in the knowledge-base; (3) use of combined model-based and rule-based reasoning; and (4) use of trend analysis heuristic rules. [3]

This research has lead to the development and use of a multi-purpose Model Toolkit (MTK) [4] and Executive Toolkit (XTK) for model-based expert systems. These tools were used to create TEXSYS, perhaps the largest real time expert system (327 rules, 349 frames, and 156,000 lines of code) to date that performs actual control of a system as well as conducting monitoring and fault diagnosis.

Specific Functionality To Be Demonstrated

TEXSYS controls the TCS in real time through the following automatic controls: analog control of the system temperature control valve, on/off control of the pump, and open/close control of 17 valves. The expert system can also call for operator assistance in performing manual functions such as heat load manipulation.

The following Nominal Operations, and FDIR elements are expected to be demonstrated in both an advisory and automatic mode:

Nominal Operations

1. Startup
2. Temperature Set Point Changes (between 35-70 degrees fahrenheit)
3. Shutdown

FDIR for ten component failures

1. Slow Leak.
2. Pump Motor Failure.
3. Single Evaporator Blockage.
4. High Coolant Sink Temperature.
5. Temperature Control Valve Failure.
6. Gas Buildup.
7. Temperature Control Valve Actuator Failure.
8. Excessive Heatload.
9. Accumulator Sensor Failure.
10. Pressure Sensor Failure.

TEXSYS provides real time control of startup, setpoint changes, shutdown, and FDIR capability for faults 2, 3, 5, and 6. TEXSYS provides passive reasoning and advice in FDIR for failures 1, 4, 7, 8, 9, and 10.

The demonstration will be accomplished by directing TEXSYS to conduct normal operations, followed by random injection of any one of the ten component faults. TEXSYS is expected to detect both the system and component level repercussions of the injected fault, and to propose a recovery technique.

Operator Interfaces

The test article status and control will be provided to the thermal engineer operator through two display screens. The "Expert System Screen" provides the operator with communication media to the expert system for control and explanations. The "Color Schematic Screen" gives the operator a "window" into the test article for information on test article status and performance. The operator can mouse on the screen to call up any system level or component level schematics he might desire for viewing in addition to data time histories.

Performance Metrics

The system performance will be evaluated as it is seen through the operator interface using the following criteria:

1. Speed and Duration: Reasoning and networking cycle times, and duration performance will be measured quantitatively.
2. Reasoning Accuracy: Accuracy of fault diagnosis and control actions will be measured relative to formally documented Operations/FDIR procedures.

3. System Robustness:System robustness to unplanned test article anomalies and hardware failures will be measured by observation and test data.
4. Flexibility: Operator interface flexibility will be evaluated subjectively by thermal engineers.
5. Displays: Operator interface display content and format will be evaluated subjectively by thermal engineers.

Project Approach

The project began with parallel development efforts in 1986 to meet a 1988 demonstration schedule. ARC began developing MTK, XTK, and operator interface tools while JSC documented its TCS expertise, built TCS brassboard hardware, and developed an integration strategy. As these early efforts neared completion, the resulting information and tools were then used in the development of control software for a TCS brassboard article at ARC.

After brassboard testing, the expert system portion of the software was modified for the prototype test article at JSC, and transferred to JSC for integration and checkout. In March 1989 testing of the integrated system was initiated using previously recorded test article data as a quasi-simulation of actual system operation. This testing continued until late June 1989, when the software was interfaced with the actual test hardware for its last seven weeks of checkout. The final demonstration is scheduled for the week of August 28, 1989.

Lessons Learned

Many valuable lessons have been learned in the course of the TEXSYS project's design, development, integration and test phases. The lessons are discussed below.

1. Specifically identify the user early in the project and focus efforts to solve his application problem. TEXSYS experienced minor problems in this area by selecting the application test article fairly far into the project, after building expert system toolkit capabilities that were not all required for the test article.

2. Real time considerations can be mitigated by choosing an application whose parameters change slowly with time and by using powerful dedicated computers. In TEXSYS, this approach eliminated most timing considerations, but careful analysis and utilization of the LMA symbolics network software was still required.

3. A new technology's operational immaturity, coupled with a lack of appropriate expert system tools adds time to the development effort. Time was invested early in the TEXSYS project to document the new application expertise and to develop the toolkits, before any real application software efforts could begin.

4. In an expert system assisted conventional control system, define clean, highly specified interfaces between the AI software system and the conventional software system. For TEXSYS, this interface took the form of a list of modular subroutines that the expert system uses to communicate with the conventional software. This approach resulted in a minimum of integration problems.

5. Iterative coding and testing the expert system software can both improve the users' understanding/acceptance of the software and improve the software's capabilities. TEXSYS was first tested against a brassboard test article, which stressed the performance aspects of the system. The software was then tested against actual test data from the application hardware, which improved the accuracy and repeatability of the system. Final testing directly on the application hardware will complete the iterative process. An alternate approach, described below, is recommended for further research.

Recommendation For Further Research

Although brassboard hardware and previously recorded test data can be used for iterative testing of expert system software, a properly designed simulation of the hardware system should be considered for this purpose and also for the development of the expert system. This simulation could actually replace the use of hardware/test data, especially during the early and mid-stages of a project, or could be used in conjunction with hardware/test data. The precise form that this simulation should take is an open research issue that should be addressed because the result may be enabling technology for the development of complex knowledge-based controllers. A development cycle that consists of 1) developing and testing the expert system using a simulation, 2) testing the expert system with the hardware and identifying properties of the simulation that are inconsistent with the hardware, and 3) correcting the simulation, and repeating 1) would be an efficient development cycle for extending conventional controller technology via the use of knowledged based systems. In this paradigm, the simulation serves, in some sense, as the repository for the current understanding of the hardware and is updated as that understanding improves. Ultimately this process may result in the simulation becoming a part of the controller software.

Conclusion

Although the SADP thermal expert system has not yet been demonstrated, it is expected to provide enhanced system capabilities and operator performance. TEXSYS is one of the

largest real time expert systems that has been implemented, and is significant in that it will perform control and fault diagnosis of a complex system. The project has been a valuable experience for the developers, integrators and domain experts, and lessons have been learned that can be put to use on future software projects. An important lesson learned was to specifically identify the end user early and have the user continuously involved in the development process. SADP reduced its real time considerations by choosing an application whose parameters change slowly with time and by using powerful dedicated computers. The new thermal system technology's operational immaturity and lack of appropriate expert system tools added time to the project's development effort. The conventional to AI software integration time was held in check by defining clear, specific interfaces. And finally, an iterative process of software development and test appears to be an effective way to produce expert system software. Further research into the use of simulation software in this process is encouraged.

References

[1] Bull, J. S., Brown, R. M. , Friedland, P., Wong, C. M., Bates, W., Healey, K. J., Marshall, P.; "NASA Systems Autonomy Demonstration Project: Development of Space Station Automation Technology"; 2nd AIAA/NASA/USAF Symposium on Automation, Robotics, and Advanced Computing for the National Space Program; Arlington, Virginia; March 9-11, 1987.

[2] Healey, K. and Hack, E., "TDAS: A Thermal Expert System Data Acquisition System," SOAR-87: 1st Annual Workshop on Space Operation, Automation and Robotics; NASA/USAF/Univ. Houston, Houston, Tx, August 5-7, 1987

[3] Glass, B. J. and Wong, C. M.; "A Knowledge-Based Approach to Identification and Adaption in Dynamical Control Systems"; 27th IEEE Conference on Decision and Control, Austin, Texas, December 7-9, 1988.

[4] Erickson, W. K. and Schwartz, M. R.; "MTK: An AI Tool for Model-Based Reasoning"; 3rd Annual Conference on Artificial Intelligence for Space Applications, NASA MSFC, Huntsville, Alabama, November 2-3, 1987.

StarPlan: A Model-based Diagnostic System for Spacecraft

Dennis Heher Paul Pownall

Artificial Intelligence Group
Ford Aerospace Corporation
1260 Crossman Avenue
Sunnyvale, California 94089

ABSTRACT

The Sunnyvale Division of Ford Aerospace has created a model-based reasoning capability for diagnosing faults in space systems. The approach employs reasoning about a model of the domain (as it is designed to operate) to explain differences between expected and actual telemetry; i.e., to identify the root cause of the discrepancy (at an appropriate level of detail) and determine necessary corrective action. A development environment, named Paragon, has been implemented to support both model-building and reasoning. The major benefit of the model-based approach is the capability for the intelligent system to handle faults that were not anticipated by a human expert.

The feasibility of this approach for diagnosing problems in a spacecraft has been demonstrated in a prototype system, named StarPlan. Reasoning modules within StarPlan detect anomalous telemetry, establish goals for returning the telemetry to nominal values, and create a command plan for attaining the goals. Before commands are implemented, their effects are simulated to assure convergence toward the goal. After the commands are issued, the telemetry is monitored to assure that the plan is successful. These features of StarPlan, along with associated concerns, issues and future directions, are discussed in this paper.

INTRODUCTION

The satellite network of the United States is a strategic resource which requires continuous monitoring and maintenance to ensure it supports defense requirements. System support personnel must carefully and precisely monitor and command individual satellites to sustain the satellite's readiness.

In current operations, when anomalies occur, a carefully developed process of evaluation, testing, diagnosis, and planning is executed by a team of highly trained engineers which support each satellite system. This process is applied incrementally to safe the vehicle, isolate the source of the problem, resolve the anomaly, and continue operations. Later, this process is permanently recorded as a contingency procedure and utilized whenever similar conditions reoccur.

Ford Aerospace Corporation, Sunnyvale Division, has been working in the field of Artificial Intelligence since the early 1980's developing a system called Paragon which, when given the proper functional description of a satellite, can monitor telemetry data, notice anomalous conditions, and recommend corrective actions.

PARAGON

Paragon is one of Ford Aerospace's innovative development environments for building model-based "intelligent" systems. It is an unusually effective software and interface system, which allows the user to go directly from idea to implementation simply by describing domain components and their behavior with logical or mathematical functions. In most cases, these can be entered simply by mouse selection within a structured window and menu driven interface. Paragon allows an expert to transfer his mental model of the domain to the computer without being taxed by normal coding and software development procedures.

Knowledge Base Development

Paragon provides automated knowledge acquisition aids that interact with an expert system developer to build a knowledge base that is a model of the problem domain. The developer is given design freedom to model a domain in a way that is most natural to his or her application.

The model consists of concepts (physical or non-physical objects) that comprise the domain, appropriate characteristics of the objects (e.g., height, weight, color, current, voltage, etc.), the interaction or relationships with other domain concepts (e.g., electrically connected to, supplied by, etc.), the behavior of the concept such as the states in which it exists (e.g., ON, OFF, IDLE, etc.), what events occur while in each state, and what causes the concept to transition from one state to another.

The model is developed via a graphic interface using pop-up menus and mouse selection. The use of typing is limited to assigning names to concepts, states, etc. Once a name has been assigned, it appears in menus or graphic displays for subsequent selection.

As the model is being developed, Paragon collects the information and automatically translates it to a representation designed for inference and problem solving. A simulator option is provided that automatically generates software code so that the behavior can be simulated and parameters displayed for verification by the system developer.

Concepts can be conceptual or physical objects (or components) that have specific meaning, relationships, and behavior in the domain. For example, in the Electrical Power Subsystem (EPS) of a satellite some of the components would be +Y WING, -Y WING, BATTERY 1, BATTERY 2, and BATTERY 3. Once the concepts are decided upon, the developer creates a classification

definition. For example, BATTERY 1, BATTERY 2, and BATTERY 3 belong to the general class named BATTERIES (see Figure 1). When classification is complete, the developer designates composition relationships. The specification of concept attributes and functional relationships follow.



Figure 1. Class and Instance classification example.

Each concept has attributes that, once defined, allow the developer to (1) localize all characteristics and behavior of an object and (2) specify functional relationships between objects. The telemetry measurements can be attributes of specific concepts which relate to components on the vehicle. For example, the attributes for the +Y WING and -Y WING would be CURRENT and SUN ORIENTATION. Any characteristics of a component or object can be specified as an attribute.

Once attributes are defined, their value class is specified. A value class designation indicates what type, or class, of values a particular attribute may take on. For example, an attribute indicating whether a component was on or off would have an ON/OFF value class type. This type would differ from the temperature of a battery, which would be a numerical value. At this point attributes can be used when specifying functional relationships between concepts and when specifying concept behavior.

Functional relationships allow the developer to specify relationships between objects or components. Figure 2 displays an example of relationships between the +Y WING and other objects in the model. The "causes" window displays values which are passed to the +Y WING and the "effects" window displays those values which are passed from the +Y WING. Each functional relationship includes a value class specification and only an attribute with the same value class as the functional relationship can be passed by that relationship. This prohibits the developer from accidentally passing, for example, an ON/OFF value when a numerical value is required.

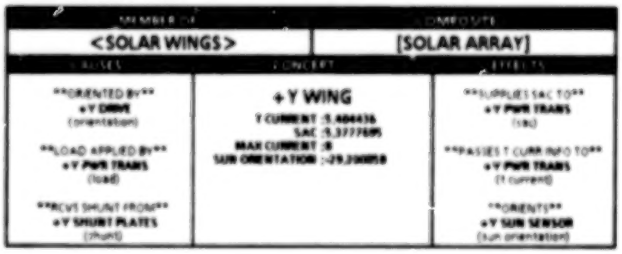


Figure 2. An example of Relationships.

Concept behavior is specified by defining (1) the states in which concepts can exist, (2) the transition conditions which determine when concepts leave one state and enter another, and (3) the attribute events which may occur in each state. Transition conditions are specified in the form of a logical operation with equations, and attribute events are specified in the form of equations.

Once concept behavior has been specified, Paragon has a "simulator" option that allows the developer to test and verify the modeled behavior. Simulations can be done at the single concept level or at the full knowledge base level. The developer is given a large amount of freedom for building simulation displays. A display can be designed that best fits the nature of the behavior to be tested or demonstrated. Display options include dials, strip charts, simple values, and flashing alarms.

Paragon's Reasoning Modules

Once a domain expert has finished building a knowledge base, Paragon can reason intelligently about the behavior as described in the knowledge base. Paragon has a collection of reasoning modules which can spot anomalous or unexpected attribute values, assess the situation and generate a list of components that could be involved with the anomaly, generate goals to correct the anomaly, and then develop a plan which will satisfy the goals.

Paragon's Data Monitoring module continually monitors the value of each attribute and when a value which is outside normal expectations is noticed, an alarm is raised. The monitoring is based upon notifications which are statements attached to concepts that specify conditions which can activate the intelligent system. Paragon's Data Monitoring module continually examines whether the current value of each attribute "matches" the defined notification condition.

Once notification occurs, the Situation Assessment module generates a ranked list of components which could have participated in causing the notification. The ranking is a "focusing" mechanism based upon the functional relationships defined within the knowledge base. With this assessment list, Paragon's reasoning modules have a significantly narrowed search space in which to find a solution to the anomaly.

With the results of the Data Monitoring module and the Situation Assessment list, the Goal Determination module identifies a change in condition (a goal or goals) which would return an out-of-limits component to nominal behavior.

Using the highest ranked component(s) identified in Situation Assessment and the goal(s) associated with that component generated from the Goal Determination module, the Planning module searches for events which have the potential to achieve the goal(s). This search is a traversal of the knowledge base across functional relationships and events that indicate, by convergence, that they would satisfy the goal(s) are identified. The transition conditions that cause these events are searched for the commands or actions which enable these events to occur.

Upon finding a plan to satisfy the given goal(s), the Planning module recommends the plan and awaits a response. If the plan is executed, the Planning module monitors the attribute values to see if indeed they do return to nominal ranges.

In order for the intelligent system to accurately confirm its operating hypothesis, the design of the knowledge base must accurately reflect the satellite command and control functionality.

The Planning module completes anomaly resolution when all goals which have been developed are achieved or, in the case of serious system failures, they cannot be achieved.

STARPLAN

StarPlan is a prototype system built with Paragon which monitors conditions onboard the Electrical Power Subsystem of a satellite, identify and diagnose problems, and advise the operator on how best to continue operations.

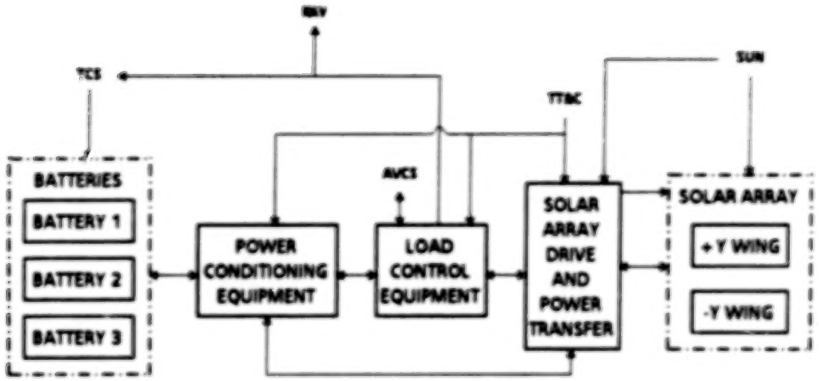


Figure 3. Functional diagram of the EPS.

The user of StarPlan would continue to control the health and status decisions concerning the satellite, but instead of asking experts to analyze the situation, the operator would simply review the recommendations of the intelligent system, making queries for additional information when necessary, and approving actions which implement the best available alternative for resolving the anomaly. With this system, the analysis, planning, and resulting command sequences are developed by StarPlan rather than by a team of satellite experts.

StarPlan Design

StarPlan consists of two knowledge bases: the first being a functional model of the EPS, and the second knowledge base a simulation model of the EPS. The functional model is a replication of the components and the relationships among those components of the EPS to provide the essential knowledge for the intelligent system to properly reason about a satellite. You could think of this knowledge base as a machine representation of a true-to-life physical model of the system.

Figure 3 is a functional diagram of the EPS and Figure 4 depicts the design of the composition of the EPS knowledge base.

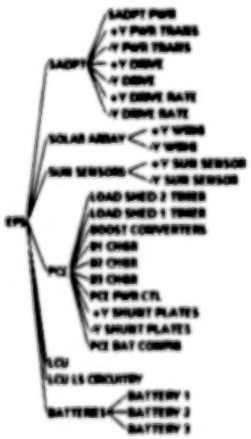


Figure 4. Composition of the EPS knowledge base.

The primary consideration in developing the knowledge base was the desire to accurately reflect the design of the actual satellite to the level of equipment configurations and functionality. When initially developing the knowledge base, the designers attempted to group too much behavior in top level components. This forced the designers to try to capture behavior that is so diverse that one did not get an intuitive "feel" for the model. The final design used subcomponents where the behavior specifications were still complex, but much more understandable.

The second knowledge base, replacing the actual satellite, is used only as a simulation model; to generate telemetry necessary to test the intelligent system. To test the intelligent system, the designers have modified this model in such a way that faults can be simulated. The faults added to this model include:

- BAD SUN SENSOR: A solar wing is unable to track the sun due to a zero error being returned by a failed sun sensor.
- WING DRIVE POWER FAILURE: A solar wing is unable to track the sun due to a System A power source failure.
- WING DRIVE ELECTROSTATIC DISCHARGE: A solar wing is unable to track the sun due to a anomalous logic change placing the wing in the hold mode.
- WING TRACKING CIRCUITRY FAILURE: A solar wing is unable to track the sun due to a tracking circuitry failure.
- BATTERY 3 THERMAL COVER DEGRADATION: Battery 3 overheats due to thermal cover degradation and a high sun incidence angle.
- BATTERY 3 HEATER THERMOSTAT FAILURE: Battery 3 overheats due to thermostat failure in the A string battery heater.
- LOAD SHED 1 TIMER FAILURE: The load shed 1 timer begins timing out independent of normal system control.

StarPlan Demonstration

The following is a description of the sequence of events during which the +Y Wing Drive Power Failure anomaly is resolved.

The satellite ground station acquires the satellite and begins to process the health and status telemetry data. Monitoring the telemetry data, StarPlan notices that several data points are out of range. Figure 5 displays the EPS telemetry data, with those values that are out of limits being highlighted.

From the notifications, the Situation Assessment module generates a list of potential components involved in the anomaly. Figure 6 displays which components could be involved with this

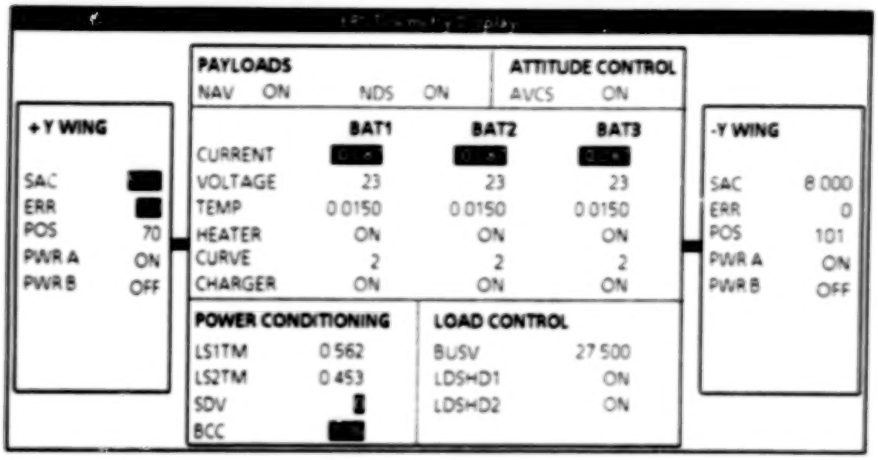


Figure 5. EPS anomalous telemetry data.

anomaly. Notice that the top ranked components are all related to the +Y WING, thus narrowing the search space for the other reasoning modules.

Rank	Concept
7	+Y DRIVE
7	+Y SUN SENSOR
7	+Y WING
7	+Y PWR TRANS
6	PCE PWR CTL
6	LCU
6	-Y DRIVE
6	-Y SUN SENSOR
6	-Y PWR TRANS
6	-Y WING
5	[illegible]
4	[illegible]
3	[illegible]
3	[illegible]
3	[illegible]
3	[illegible]
3	[illegible]
3	[illegible]
1	[illegible]

Figure 6. Situation Assessment List.

The Goals display is shown in Figure 7. The top two goals (there are actually seven goals, but only the top two are shown) are related to the highest ranked components in the assessment list. The goals, displayed in an English-like syntax for easy understanding, are essentially saying that the +Y WING needs to be rotated. But the Planning module has to figure out *how* to rotate the wing.

Goals

1) The value of the SAC attribute of the +Y PWR TRANS component is greater than 5.3 AND the value of the SAC attribute of the +Y PWR TRANS component is less than 8.9
2) The value of the SUN ERROR attribute of the +Y DRIVE component is greater than -1.5 AND the value of the SUN ERROR attribute of the +Y DRIVE component is less than 1.5

Figure 7. The Goals display.

The Planning module takes the top ranked goals and tries to find a course of action or actions that would satisfy the goals. The Planning module searches the knowledge base for events that indicate they would satisfy the goals. This is found by simulating the events looking for a trend that indicates a convergence to satisfying the goals. Once an event, or a series of events are found which could satisfy the goals, the Planning module determines what commands sent to the satellite would cause these events to occur.

Throughout the knowledge base, commanding information is "embedded" in the transition conditions for various components. The embedding of commands in transition conditions enabled the Planning module to locate commands which can potentially change the anomalous behavior of the satellite back to normal.

Once a commanding plan is found, but prior to sending any command to the satellite, the plan is verified using the knowledge base behavior specifications to validate that the anomalous conditions will be improved. Their effects are verified internally using the knowledge base specifications to confirm their effect on the satellite prior to their actual use in commanding. The Planning module is then able to determine whether to try another approach or to verify that the present planned approach is achieving the intended goals.

Once a commanding plan is verified via the knowledge base, commands are sent to the satellite (in StarPlan they are sent to the simulation knowledge base) to gather more information about the anomaly by monitoring its subsequent behavior. This process is designed to "safe" the vehicle while testing the expert system's current operating hypothesis concerning the resolution of the anomaly.

The first command found is to put the +Y WING into track mode using power system A. This command is sent, and the StarPlan monitors the telemetry data for a response. After waiting for a short while, StarPlan realizes that the track command is not working. The next command is to manually rotate the +Y WING. Once again, after waiting a short while StarPlan realizes that this command is also not working. The third command to try is the track command but with power system B. StarPlan notices that power system B is not currently on, so the command to turn it on is sent. Once power system B is on, the track command is sent. This command works (the wing position starts increasing) and StarPlan monitors the telemetry data until all of the goals are satisfied and the telemetry values return to normal (Figure 8).

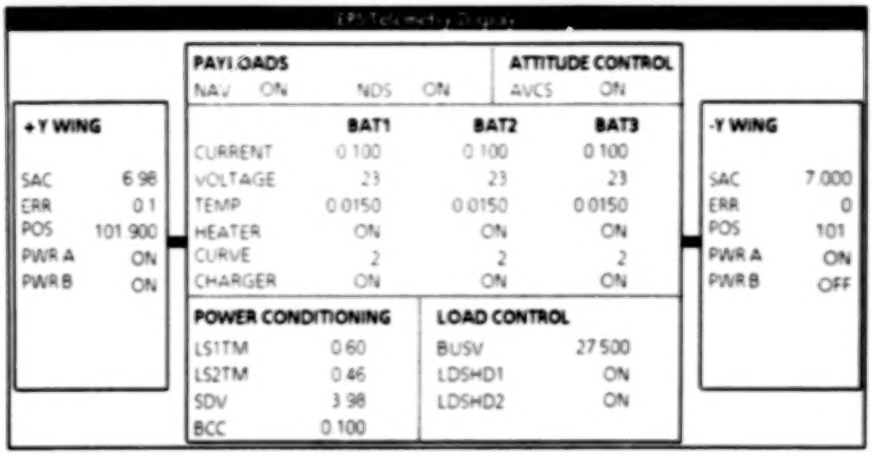


Figure 8. Normal EPS telemetry data.

CONCLUSION

Paragon is an easy to use system to build accurate functional models of a domain, such as satellites, combined with a collection of reasoning modules that use the model to resolve anomalies. Most importantly, the anomalies resolved can be completely unanticipated by human experts. The model built can be at any level of complexity, however, the more detailed the models, the finer the resolution of anomalies.

The reasoning modules described here are still being developed. As new issues arise and more complicated anomalies are tested, further enhancements or corrections become necessary. We feel confident that our reasoning approach will be able to handle many difficult to solve anomalies.

StarPlan is a prototype expert system that can handle faults on board a satellite, with only the Electrical Power Subsystem currently being modeled. Numerous anomalies have been tested with StarPlan, all of which have been resolved correctly. Further extensions to StarPlan are expected, with a complete functional model of a satellite being our ultimate goal.

REFERENCES

1. Blasdel, Arthur, "Automated Fault Handling of a Satellite Electrical Power Subsystem Using a Model-Based Expert System", Ford Aerospace Corporation internal report, Sunnyvale, California, 1988.

2. Ford Aerospace Corporation internal report, "The Paragon Development System User's Guide", Sunnyvale, California, November, 1988.

3. Hendler, James A., *Integrating Marker-Passing and Problem Solving*, Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, 1988.

4. Sathi, Arvind; Fox, Mark; Greenberg Michael, "Representation of Activity Knowledge for Project Management", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, Vol. PAMI-7, No. 5, September, 1985, pp. 531-552.

5. Wilensky, Robert, *Planning and Understanding: A Computational Approach to Human Understanding*, Addison-Wesley, Reading, MA., 1983.

SMART BUILT-IN TEST

Dale W. Richards
Rome Air Development Center
Griffiss AFB NY

ABSTRACT

The work which built-in test (BIT) is asked to perform in today's electronic systems increases with every insertion of new technology or introduction of tighter performance criteria. Yet the basic purpose remains unchanged -- to determine with high confidence the operational capability of that equipment. Achievement of this level of BIT performance requires the management and assimilation of a large amount of data, both realtime and historical. Smart BIT has taken advantage of advanced techniques from the field of artificial intelligence (AI) in order to meet these demands. The Smart BIT approach enhances traditional functional BIT by utilizing AI techniques to incorporate environmental stress data, temporal BIT information and maintenance data, and realtime BIT reports into an integrated test methodology for increased BIT effectiveness and confidence levels. Future research in this area will incorporate onboard fault-logging of BIT output, stress data and Smart BIT decision criteria in support of a singular, integrated and complete test and maintenance capability. The state of this research is described along with a discussion of directions for future development.

Introduction

The approach to maintenance of electronics has for many years been narrowly scoped and highly segregated. Systems and subsystems located on an operational platform are identified as having failed or contributed to a failure. These units are removed and sent to a maintenance facility where the process is repeated with circuit cards being separated from the boxes and sent to another facility for removal and replacement of components. This decomposition follows naturally from the hierarchial manner in which electronics have been designed and built. Each maintenance step is performed independent of the others with items designated as "failed" at one step removed from the parent unit and sent to the next step of maintenance, often with little or no supporting data as to why that item was designated as "failed" in the first place. Though performing well to date, this approach to maintenance has fallen short in recent times as the number of maintenance actions has risen to levels severely taxing the logistics and support resources provided. Compounding this situation is the increased complexity of those systems and the cost of their maintenance. In addition, the nature of many missions precludes a fixed maintenance facility, requiring that extensive maintenance resources be deployed along with the equipment itself. The current technology is able to meet the immediate needs of the logistics and support community, but provides no inherent response to the concerns of maintenance and diagnostics, thus impacting the long-term needs of that community. The level of false removals, as indicated by high cannot duplicate and retest okay rates, sometimes in excess of 50%, which are associated with many currently fielded systems prevents the achievement of acceptable levels of availability. Any modification to on-board testing for these systems must properly reflect the increased need for even more efficient operation. Every attempt must be made to minimize the degree to which good units are removed from the operational platform and to maximize the extent to which knowledge concerning the equipment failure and subsequent designation of subunits contributing to that failure is effectively conveyed to subsequent maintenance and therein utilized.

Achievement of this level of performance requires an increase in the ability of built-in test (BIT) to properly distinguish between hard failures, intermittent behavior and one-time false alarms. Critical to this process is the correlation between the BIT output and knowledge of the physical environment incident to the perceived failure. This alone can only reduce the number of units entering the maintenance pipeline, but to reduce the incidences of cannot duplicates and retest okays it is necessary that relevant data concerning the declaration of a "faulty" unit be automatically included with that unit as it is removed from the platform and sent on for further maintenance. Some form of non-volatile electronic storage is required along with corresponding capabilities in the associated automatic test equipment (ATE) to

recover this data and effectively utilize it in the diagnostic and failure confirmation process.

RADC Response

Rome Air Development Center (RADC) has sponsored a number of research efforts directed at developing technologies in support of the above capabilities. This work has been concentrated at the system/board/module level. There are two components to this research. The first and principle aspect of this research is concerned with decreasing the number of false removals which contribute significantly to the infamous false alarm/cannot duplicate/retest okay problem and increasing the ability to identify intermittent failures. The building blocks for this work are techniques developed through research in artificial intelligence (AI) and the approach is collectively known as Smart Built-in Test (Smart BIT). The second and supporting aspect is concerned with the measurement, recording and correlation of environmental stresses such as vibration, shock and temperature along with quality of the prime power supply. The goal of this work is the development and integration of a single micro-electronic package known as a Micro Time Stress Measurement Device (TSMD) which could be located within a removable/repairable unit.

Smart Built-in Test

A major problem facing today's maintenance community is that of high false alarm rates. Units removed at one level of maintenance often test good at the next level. This leads to inefficient use of personnel, test equipment and spares and can contribute to lessened availability of equipment and increased demands on the logistics system, paricularly when whole systems are deployed to areas without preexisting maintenance facilities. A number of approaches have been suggested for improving the efficiency of this process and targeted at every corner of the maintenance concept picture. However, the greatest impact is achieved when the front end of the process is improved -- reduce the number of false removals! This is the approach taken by Smart BIT. Smart BIT is best thought of as an adjunct to the actual functional test performed by traditional BIT. In its current form it is a software filter on the output of the functional test but it could easily be integrated as a singular BIT function. Current BIT technology often places 100% confidence on the results of a test, even though these results could be biased by the behavior of other units or temporally influenced by transient environmental conditions. Incorporation of an N-out-of-M filter can improve the condition to some degree, yet even it can be easily misled. Smart BIT goes beyond these simplistic approaches to include a more robust reasoning process that looks for information in the pattern of faults and incorporates knowledge of time and other information outside of the functional realm of BIT.

Research to date has centered around the application of Smart BIT techniques to two separate systems containing BIT. The first of these was a signal-conditioning and reporting module utilizing primarily discrete components and low levels of integration. The second was an air-data computer utilizing highly integrated ICs and a bus oriented architecture. Different techniques were developed for each system and then applied to a laboratory simulation of those systems. A common goal during the development of the techniques was to maintain a sufficient level of generality to ensure that the technique could be applied to other systems without starting over from the beginning. Thus an implementation of Smart BIT will involve the selection, ranking and integration of multiple Smart BIT techniques based on the BIT and mission needs of system being modified. The principal techniques identified for which software has been developed and demonstrated aree Information Enhanced BIT, Improved Decision Rule BIT, Temporal Monitoring BIT and Adaptive BIT.

Information Enhanced BIT was a precursor to the concept of integrating TSMD with Smart BIT. BIT decisions are based on information internal to the unit under test (UUT) as well as other, external sources. These could be environmental monitors or information concerning the operational mode of the platform or the health of other systems. In general, the output of the BIT is compared against known failure modes and the external data used to corroborate any fault identifications.

Improved Decision Rule BIT incorporates a structure suggestive of an expert systems format to increase the robustness of the BIT decision process. A simple BIT check such as "IF test-1 fails THEN report unit-12 faulty." could be augmented to be "IF test-1 fails AND unit-12 has exhibited intermittency AND stresses have been near threshold THEN declare unit-12 marginally healthy AND adjust sensing frequency." The important characteristics in this technique is that decisions are based not on a single fact, but that compound antecedants are often employed and any assumptions about the system that could influence the firing of a rule must be additionally substantiated.

Temporal Monitoring BIT uses Markov modeling techniques combined with a finite state machine representation of unit health to monitor performance over time. The transition from OK to HARD is forced to cross a state of intermittency. The probabilities of going between the substates of recovering and faulty are dynamic and are adjusted according to the pattern of GO/NO-GO's coming from the BIT. If the chance of recovering becomes very high, subsequent NO-GOs can be treated as intermittents and the unit declared functional but degraded.

Adaptive BIT makes use of two general learning paradigms: k-nearest neighbor and neural network back-propagation. In both cases the BIT report in question is plotted into an n-space defined by the various parameters of interest,

such as vibration, GO/NO-GO, airspeed, duration of failure, etc. In k-nearest neighbor, the k previous values plotted which are closest in absolute distance from the new point are compared with the new point entered as a GO or No-GO depending on the GO/NO-GO value of those k points. The approach used in neural nets is consistant with accepted neural net theory and in essence divides the earlier defined n-space into a number of regions, each classified as either GO or NO-GO.

Time-Stress Measurement Devices

The relationship between accumulated stresses and failure modes of equipment has long been recognized. Currently, there is no correlation between these entities — when failures occur only the effect of stresses on the equipment, the failure iself, is captured and not the actual stress conditions to which it was exposed and which may have precipitated its failing. The reason is that the ability to measure these stresses has until now been limited to the placement of a discrete transducer at the point of interest. Advances in sensor fabrication and integration, coupled with general increases in computational density now allow for a complete stress measurement and recording system to be fabricated in under two square inches. This is the purpose of research undertaken by RADC as part of its Time Stress Measurement Device work.

Stresses affecting electronic equipment can include thermal, vibration, shock, and electromagnetic signals. The variety in which these stresses exhibit themselves can range from simple discrete events to the cumulative effect of many events over a period of time. It is important to be selective about what stress characteristics to measure and what data to store for future use. This need for a measurement capability has led to the construction of a paperback novel-sized TSMD module suitable for a flight data collection program. Stresses measured include temperature, vibration/shock, and prime power quality. Data collected from both A-7 and A-10 aircraft is now being analyzed and correlated with relevant maintenance actions. A critical portion of this effort was the determination of appropriate methods of data compression as it became impractical to store a complete electronic "strip-chart" of each sensor output. To do this, many parameters are characterized by either cumulative time above a threshold or number of excursions above a threshold.

Following on the successful development and testing of the TSMD module effort is the on-going development of a micro-TSMD package to incorporate the capabilities of the module in the aforementioned two-square-inch hybrid chip. Currently at the advanced-development-model stage, this implementation will be amenable to mounting on cards in line-replaceable units or modules. Full-scale development of a qualified micro-TSMD will begin in late FY89 with availability projected for FY91. The first insertion of the micro-TSMD into an operational system will occur as part of a Warner Robins Air Logistics Center Microcircuit Technology in Logistics Application (MITLA) project related to autoidentification technology for printed circuit boards.

Technology Insertion

It still remains for Smart BIT and TSMD technologies to be integrated and then inserted into a fielded system. Steps toward that end include additional research into the integration of the various technical components and a change to the overall maintenance concept as regards the management of maintenance related data. Central to this entire process is the recording of environmental, diagnostic and logistics data local to the level at which units are removed from the operational platform, be they boxes, Line Replaceable Units, Line Replaceable Modules or some other similar designation. Current TSMD technology has the capability to store some logistics data along with a compressed record of stress data. An audit trail of the diagnostic process resulting from Smart BIT should also be stored at that level, depending on memory constraints. Maintenance actions involving the removed unit would be able to access this data. ATE could be programmed to incorporate the TSMD and Smart BIT data into its own diagnostic process and the logistics data could then be appropriately updated.

Research is now underway to define the degree to which Smart BIT and TSMD need to share information and to identify pertinent characteristics of that data to be retained in memory. It is apparent that stress data should be available at three levels of temporal resolution: uncompressed in the temporal vacinity of a possible failure, compressed for duration of a mission and statistically characterized for all such similar equipment and missions. The most appropriate means for converting realtime sensor data into these time compressed formats remain to be determined. Additionally, a proper path will be identified for transitioning the integrated technology from the laboratory into the field. The distribution and sensitivities of the stress sensing elements need to be defined and methods of icreasing BIT processing capabilities to perform additional reasoning functions determined. Not only is impractical to place a TSMD module in every circuit and a LISP machine into every BIT equipment — it isn't necessary. The relationship between a component or board in question and a remotely mounted sensor can be analytically calculated and the software to perfron the reasoning can be written in a variety of languages. The development of powerful symbolic processors and greater density memories will provide the computational processing capabilities required. The more prospective approaches being considered involve the upgrade of a specific card/module for a particular system. This would have a minimum impact on the rest of the system and the upgrade itself could result in the necessary electronic real estate for the new functions.

Further Development

The benefits of Smart BIT can best be understood from the perspective of the overall maintenance and diagnostics process. A typical maintenance scenario involving full integration of Smart BIT and TSMD capabilities will now be described.

During a mission the TSMD portion is continually recording stress profiles in a wraparound fashion, replacing old data with more recent measurements. The older values being data-compressed and stored in long term memory. The TSMD will also detect specific stress profiles that could damage equipment and note their occurrences in the long term memory. When stress data is needed by either Smart BIT or maintenance equipment, this information is retrieved from the long term memory. When a failure condition is detected by the Smart BIT the TSMD is asked to return relevant stress data. Depending on the criticality of the system to the mission and flight safety, the Smart BIT will continue to analyze both the functional test data stream and the TSMD output. If necessary, this process may be performed offline while a spare unit is switched in place of the one in question. If a decision is made to declare a unit faulty, information relevant to that decision process will be stored local to that unit's non-volatile memory for access later by other maintenance processes. At the flight line, units identified as possibly faulty will make themselves known to either human personnel or directly to ATE. Maintenance at this stage will remain primarily a remove and replace process, excepting simple procedures such as tightening, aligning or connector replacement. The critical distinction is that the units removed have with them relevant diagnostic and logistic data. During subsequent maintenance action for those units ATE software will request from the non-volatile memory the information placed their by the TSMD, Smart BIT and previous maintenance cycles. This is data that the ATE has no other means of attaining and will be used to discriminate among fault isolation choices, guide the direction of diagnosis or suggest candidates when no failure is indicated by the ATE functional tests. Periodically, information collected from many maintenance actions is analyzed, and revision and data updates forwarded to the field for loading into the Smart BIT and TSMD software, or modifications may be made to the logistics data stored within the unit-under-test itself.

Conclusions

The future direction for maintenance points toward a greater role for on-board testing and greater data interchange between stages of maintenance. The first can be provided by the inclusion of Smart BIT capabilities. They, in turn, will generate new and more confident data regarding the failure or operational status of the equipment being tested by the BIT. This alone will reduce the number of healthy units unnecessarily removed from a platform. And, in conjunction with an implemented imbedded logistics data tracking system, can serve to reduce the burden on already scarce maintenance resources. Only with a streamlined maintenance process wherein every action is a necessary one and resources are utilized to optimal efficiency can the R&M goals of greater availability and reduced logistics support costs be met.

References

Zbytniewski, J. D., Anderson, K., "Smart BIT-2: Adding Intelligence to Built-in Test," NAECON '89, May 1989, pp 2035-2042.

Collins, J. A., "Time Stress Measurement Device: A Tool for BIT," Proceedings of ATE & Instrumentation Conference-East, June 1989, pp 1-8.

Solar Array Automation Limitations

Terry M. Trumble
Aero Propulsion and Power Laboratory
Wright Patterson AFB 45433-6563

ABSTRACT

Significant progress in the automation of the spacecraft electrical power systems has been made within the past few years. This is especially important with the development of the space station and the increasing demand on the electrical power systems for future satellites. The key element of the spacecraft power system, the solar arrays which supply the power, will have to grow to supply many tens of kilowatts of power within the next twenty years. This growth will be accompanied by the problems associated with large distributed power systems. This paper addresses the growth of the arrays, the on-array management problems and potential solutions to array degradation or failure. This paper will be primarily limited to the discussion of multilowatt arrays for unmanned spacecraft with comments of the implications of array degradation for manned spacecraft.

1. Introduction

The electrical power system (EPS) requirements for large unmanned satellites are considerably different from those of manned satellites. The large size of manned spacecraft, such as the space station, requires considerable electrical power to ensure a long, useful mission. Maintenance of the health of the system is provided by the combination of expert systems and human judgment. The envelop of possibilities for maintaining the electrical power system using this combination of man and machine are almost unlimited. Determination of the abrupt degradation of the array can be as simple as visual assessment, or as complex as a computer analysis of the electrical system performance correlated with orbit location and maneuvering constraints. All of the options invariably include the human in the loop.

At the present time, for the unmanned spacecraft, human prognosis and intervention in the case of an EPS anomaly, is limited by the information provided about the array and the physical limitations of communicating with the spacecraft from a number of ground stations. Additionally, the need for an expert at the ground station becomes obvious when the problem presented by the EPS does not lend itself to straightforward solutions. Within the past few years, ground based experts have become expert systems. They act more rapidly and reliably than their human counterparts, however they have to wait for the communication window between the satellite and the ground station. Additionally, they are based upon past measured performance with the identification of specific anomalies and correction or compensation procedures. This type of information is available for the EPS but is not generally available for solar arrays. Solar arrays themselves are almost devoid of instrumentation and are basically, fixed configuration, electrical power generating systems. In order for them to survive on an unmanned spacecraft, they must have the same advantages as those provided for manned spacecraft. The man-in-the-loop must be replaced fully by a sensing, assessing, problem solving entity. In short, the array itself must take on human characteristics.

2. Solar Array Considerations

The solar arrays under consideration are planar, oriented arrays as opposed to spacecraft body mounted arrays or solar concentrator arrays. A 10 kilowatt array of silicon solar cells will require about 61 square meters of area. This can be broken down into several smaller arrays, typically four arrays of slightly over 15 square meters. Conventional arrays on the order of 10 to 15 kilowatts have already been built and tested in the support of manned spacecraft. The problems encountered were dealt with in a reactive fashion rather than a proactive fashion. The short, limited lifetime of these arrays has not been a major problem in the past. With the requirement to last 10 years, unattended, arrays of this size require on-array instrumentation, processing and communications with the main EPS computer.

The new breed of solar cells for planar arrays have reached the 20% efficiency plateau. This reduces the 10 kilowatt array size from 61 square meters to 42 square meters, a reduction of over 30 percent. In the 21st century, spacecraft requirements can be expected to grow to 50 or 100 kilowatts. In the worst case scenario, using ultralight silicon solar cells 100 kilowatts would occupy 420 square meters. How can the problem of maintaining adequate power from such a large array for over a decade be accomplished

without a human performing extra vehicular activity (EVA)? The solution lies in the design of a self monitoring array with the capability to reconfigure itself.

3. Design Philosophy

Selection of the array voltage is a critical parameter that must be considered before all other considerations. The selection of the voltage is orbit dependent and beyond the scope of this paper. The selection of 160 volts direct current (VDC) however, has been chosen as a practical voltage that is applicable to different spacecraft orbits. Configuring the array into blocks of solar cells to produce 160 VDC can most easily be accomplished by putting 185 gallium arsenide (GaAs) solar cells in series, and paralleling at least five series strings. A block of 925 2 cm X 2 cm solar cells then provides 100 watts of power, with a total of 1000 blocks required for the 100 kilowatt array.

Using these numbers, it is now possible to discuss some of the requirements and alternatives available to ensure ten years of uninterrupted power from the array. An obvious improvement can be made if one poorly performing series string of 185 GaAs solar cells could be dropped off-line and replaced by a properly functioning cell string. This would require 5,000 switching transistors, one for each of the cell strings. The 5000 switching transistors would replace the diode isolators used for each cell string. Each transistor would be required to switch 20 watts. Each transistor could be designed so that it could communicate through the B+ lines. I could be used as a current sensor which would allow it to provide this information to a local microprocessor responsible for comparing current from all the switches. The new string would automatically go on-line through either a command from a local microprocessor tracking the number of cell strings on-line, or automatically as a result of the transistor providing the low current information.

On-array sensing (Figure 1) must be accomplished if array integrity is to be maintained. Voltage sensing can be for the entire array and therefore is rather straightforward. Current sensing should make it possible to identify each series string. Another method to accomplish this task would be to measure the current going through a string by turning a string off then back on. The value of current could be compared to a stored standard. Current (I) and voltage (V), however, are not the only parameters of interest. The cell string temperature is important, however the penalty for embedding large numbers of temperature sensors in the array to sense the temperature of each cell string would result in at least 5000 sensors with the accompanying communication and conversion requirements. A minimum of one sensor for each segment of the array is necessary. Temperature sensor data from one block can be compared with temperature sensor data from neighboring blocks and a built in reference standard. This allows trends and out of tolerance temperatures to be identified. Predictive solutions can be used if enough historical information can be stored.

Other sensors will also be necessary. For a large array, it is important that accelerometers and vibration pickups be used to provide information describing the mechanical state of the array. Collisions with space debris, orbital corrections, and mechanical failures have to be detected and identified if compensation or correction is to be meaningful. A critical issue that requires clarification is the issue of correction versus compensation. An intelligent array is primarily oriented toward the correction of problems, with compensation as a secondary solution. To correct array problems, the array must be electrically reconfigurable and must have other attributes that make it worth while to extend the lifetime of the array. Annealing gallium arsenide solar cells on the array is one method for correcting a major array problem. Annealing, raising the temperature of the array until solar cell radiation damage has been appreciably reduced, can be accomplished a number of ways. It is conceivable, for example, to pass electrical current through a string of solar cells in the forward direction until the heat generated is adequate to generate annealing. For a large array, it is possible to use several active solar cell strings to forward bias another cell string until it has recuperated 85 to 90 percent of it's Beginning-of Life (BOL) power. This cell string can in turn be used to anneal other cell strings.

4. Smart Sensors

The problem of sensors for a large array creates a completely new concept for the array design. It is totally impractical to develop a sensor that requires a power supply to operate, a separate amplifier and power conditioner, and miles of shielded cable. The alternatives are quite

obvious. Smart sensors must be developed. A smart sensor, for this case, is described as an integral sensor/microprocessor package that operates off the solar array B+ voltage and uses the electrical wiring for both communicating with the array processors and providing sensor information. With this concept, embedded sensors make sense and weight penalties become negligible. On-array communication between the main on-array microprocessor and microprocessors distributed across the array will require a handshake or token protocol to ensure that each microprocessor is operating properly. If the main computer on-board the spacecraft fails to query the on-array main microprocessor, this microprocessor would query the main computer to determine its health. Failure of the main computer to respond would then allow the main on-array microprocessor to fully control the array with reporting still being continued to the main computer as a matter of routine. The communication between the smart sensors and the on-array computer would require that the smart sensor have a priority token. When there is any change in the status-quo for the sensor, it sends out its priority token with the sensor information. As the information is received the priority arrives with it to allow the microprocessor to select the most important information. The details of operation of such a system can be worked out by several methods, but will not be discussed in this paper.

5. Sensor Information

The basic rule for the use of sensor generated information is to minimize it. Smart sensors will not provide output data unless there is a status change. This step will minimize the inputs to the control microprocessors. The second step is to use as few sensors in combination as possible to generate an action. Use of accelerometer data and vibration pickup can be algorithmically combined to indicate a mechanical collision on a specific array segment. This information combined with the current sensor information for that segment provides high confidence that a specific number of cell strings are affected and that switching in of spare cell strings should be done. After array compensation has been completed, further analysis of the damage can be accomplished using other available sensors. The temporal aspects of the problem thus become amenable to simple solutions. In most cases, solutions can be accomplished rather slowly, that is, in a matter of seconds or minutes. Microprocessor speeds therefore need not be exceptionally fast, however they must be insensitive to the radiation and electromagnetic environment of space.

6. Design Philosophy

Before serious consideration can be given to the specific design of on-array hardware, rules must be formulated for both the array and the autonomous array management system. Rules for the EPS have already been formulated for manned spacecraft systems. This establishes the basis for the interaction between the EPS within the spacecraft and the autonomous array management system, even though these rules must be modified for the unmanned spacecraft case. A simple list is provided to indicate the genesis of these rules.

a. Reliability is the most important criteria.

b. Fail safe design required. The array should never degrade worse than it would without and autonomous system to manage it.

c. Minimize components count.

d. Minimize weight.

e. Design for long term performance.

f. Minimize on-array data taking, processing and storage.

g. Sense and verify critical data.

h. Establish a temporal basis for data analysis.

i. Transmit only new information from sensors.

j. Verify sensor integrity on a continual basis if practical.

These rules are provided as a basis from which to develop a complete design philosophy for the autonomous array operation and establish the basis for rules in a rule based on-array expert system. The expert system will most likely not be of a straightforward forward or backward chaining design.

7. Expert System

With the progress that has been made in the aerospace industry in the development of expert systems for the control and monitoring of the spacecraft EPS, much of the work for a fully autonomous EPS and an autonomous solar array has been completed. The present systems require a man in the loop, however there are several direct

techniques for replacing the man. The concept of power management, in many cases provides the crewmember with data after the fact. With the appropriate action already taken, this process is fully autonomous. This is generally applied to load sharing, shedding and prioritization, a major portion of the main EPS power management activity. The on-array expert system has to work closely with the spacecraft EPS computer. This interface and its requirements represent a large complex problem requiring a lengthy detailed analysis. The goal is to establish the on-array main microprocessor as an independent entity requiring minimum memory and microprocessor code. Although ADA and FORTH are proposed as candidates for the on-array language, the selection on the final language must be based upon the self maintenance of the array and its reliable function. Since there will be no maintenance of the system once it is in space all other criteria for the selection and use of a language fall by the wayside.

8. Summary

This paper is intended to provide insight into the design problems associated with a large fully autonomous array. Most of the major problems have been discussed in a fashion that will allow the reader to approach the problem with his own artificial intelligence tools. Most importantly, the paper is designed to provide the basis for the development of a sound philosophy for the development of an autonomous array. The development of an autonomous array should be considered only after alternatives such as Extra Vehicular Activity (EVA) have been considered. If it is feasible to accomplish on-site repairs in space, the design of the array will have to be modular and easy to repair. Safety of the person doing the EVA then becomes a concern because of the space environment and the array voltages. Regardless, whether a fully autonomous array becomes necessary or not, the hardware and software should be considered for development to ensure the operation of large unattended arrays both for spacecraft and for surfaces of other planets or moons. Solar array systems must be smarter if they are to survive.

List of References

1. R. J. Speir, M. E. Lifering, " Real-time Expert Systems for Advanced Power Control". Proceedings of the 23rd Intersociety Energy Conversion Engineering Conference. August 1988.

2. T.M. Trumble, "Autonomous Solar Arrays for the Future", Society of Automotive Engineers, Technical Paper Series, Aerospace Congress and Exposition. October 1984.

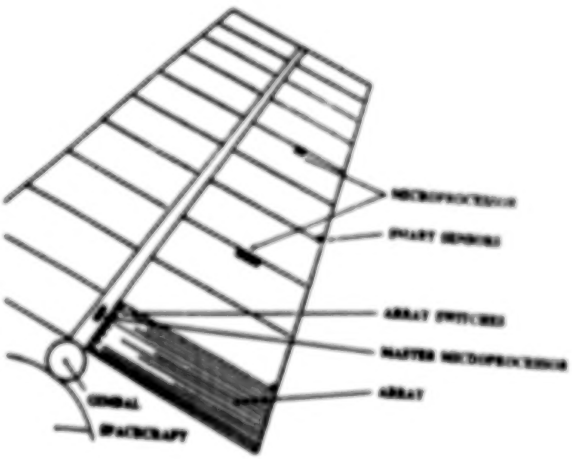


FIGURE 1

A PERSPECTIVE ON ROBOTIC RESEARCH PROGRAMS

Dr. M. Montemerlo
NASA Headquarters

(Paper not provided by publication date.)

SDIO ROBOTICS IN SPACE APPLICATIONS

BY RICHARD ILIFF

ABSTRACT - ***This paper addresses how SDIO/S/PL views robotics in space supporting the Strategic Defense System (SDS) program. It addresses ongoing initiatives which are intended to establish an initial Robotics in Space capability. This is specifically being referred to as the Satellite Servicing System (SSS). This system is based on the NASA Orbital Maneuvering Vehicle (OMV) with a Robotic*** **Manipulator(s) based on the NASA Flight Telerobotic Servicer (FTS) and other SSS equipment required to do the satellite servicing work attached to the OMV. The paper also addresses specific SDIO Robotics in Space Requirements which have resulted from the completion of the SDIO/S/PL Robotics Requirements Study Contract.**

BACKGROUND

In 1987 studies were completed which looked at Space Assembly, Maintenance and Servicing (SAMS). The purpose of these studies was to investigate ways in which satellites could be maintained on-orbit. Design Reference Missions (DRMs) were developed by NASA and the AF to be representative of satellite constellations which might exist in the mid to late 1990s. Using these DRMs the contractors developed what they considered would be the best design approach to develop an on-orbit servicing system. Along with this, design concepts were investigated to determine how the satellites themselves would have to be designed and built in order to make them serviceable. Also tools, interfaces and other on-orbit servicing design needs and requirements were investigated and recommended.

Phase I of the Spacecraft Partitioning and Interface Standardization (SPIS) study was completed in 1987 for the Air Force which looked at the spacecraft sub-system designs which might be form, fit and function compatible across different types of satellites. The recommendations for sub-system standardization which resulted from this study were; battery, power control unit, inertial reference unit, reaction wheel, earth sensor and the sun sensor. Under Phase II of the SPIS the contractor is required to develop final specifications for these ORUs. The battery standard is presently nearing approval and the power conditioning unit standardization process is under way. These standardization efforts are similar to the avionics efforts initiated by the PAVE PILLAR and Modular Avionics System Architecture (MASA) programs for aircraft. Also required within this standardization process is a separate activity to define a standard spacecraft data bus, power bus and serviceability interfaces. These initiatives will then open the way to designing satellites so that they can be built in a modular and serviceable fashion.

Following the SAMS and SPIS studies the SDIO, AF and NASA began a dialog on establishing a Satellite Servicer System (SSS) which could be ready for use on-orbit in the mid to late 1990's. These discussions eventually resulted in a project based on the Orbital Maneuvering Vehicle (OMV). The concept which evolved envisioned an OMV providing the basic servicer platform. To the OMV would be attached an Orbital Replaceable Unit (ORU) and servicing carrier with a robotic front end. This robotic front end is not yet designed but will be derived from the NASA FTS program. Thus the manipulator will use components of the FTS, although it may take on a different form than the FTS, rather than becoming a new development effort. The SSS is intended to be used in a non-man tended mode when used to maintain satellites. In the case of the SDIO which has no man-in-space requirement this robotic design takes on the requirement that it must be semi-autonomous and this requirement is sometimes referred to as supervised autonomy. Supervised autonomy implies that the robot will be able to perform some tasks autonomously but will

stop at predetermined points in the specific autonomous task being performed or will stop when the robot detects a non-programmed problem or interference. An operator will then have to interact with lar advantages through the use of robotics. The contractor was to accomplish this through five sub-tasks. The first sub-task was to do an overall robotics assessment of the state-of-the-art and the present shortfalls of the technology. Another task was to develop a Robotics Requirements Document for all aspects of the SDIO robotics program. This would lead into a Time-phased Implementation Plan and draft Program Management Agreements (PMA) for accomplishing the SDIO Robotics Program developed by the contractor. Finally the contractor was to develop a Robotics Video which would provide a quick way to educate personnel unfamiliar with robotics as to what the technology is, where the state-of-the-art is and what advancements are required to implement the SDIO program.

Robotics in space requirements - The main requirement for SDIO Robotics in Space will be for the robot to be robust. If the design constraints on the robot are too stringent then it will require a new robot to be designed for each application. The candidate on-orbit support missions that the robotics contractor included in this study were as follows:

1. Fuel transfer
2. Orbital Replaceable Unit (ORU) changeout
3. Counter tumbling satellite approaches
4. Uncooperative satellite retrieval/removal from orbit or neutralization
5. Ad Hoc tasks (using tools)
6. Ad Hoc and programmed task control
7. On-orbit inspection and calibration
8. Support of SSS components on-orbit such as an Orbiting Support Platform
9. On-orbit assembly
10. Reboost (Correction of Orbit/De-orbit)
11. Removable orbit insertion motor
12. Hazardous debris removal
13. Intercept vehicle reload
14. Nuclear reactor removal

In addition to the candidate missions above was added the requirement that the robot be capable of servicing any of the orbiting SDS space assets in both the near the robot to guide it on through the task or around the detected problem.

This supervised autonomy is necessitated by SDIO and AF satellites which are in orbits which could, because of the combination of distance and electronic or mechanical processing, have communication delays in the neighborhood of several seconds. These delays are difficult to learn how to handle by an earth based operator in a totally teleoperated mode and necessitate that the entire system be slowed down to ensure safety to the robot and the satellite. Therefore in the interest of economy of time and resources it will be necessary to allow the robot to do certain repetitive functions autonomously at a higher speed with the operator teleoperated slowdown only being required when absolutely necessary.

At the start up of the negotiations with NASA on SSS the SDIO realized that its requirements might be more restrictive than NASA's due to the supervised autonomy requirement as well as the relatively more robust satellite environment. Also the need for teleoperation from the ground rather than from on-orbit in the Space Station or the Space Shuttle implies that the SDIO and AF requirements for the servicer be more restrictive. This led the SDIO to contract for a study to define the SDIO requirements for a SSS robot. In addition this study developed robotic requirements for ground and manufacturing in relation to the SDIO Strategic Defense System (SDS) and its followon components. However only the requirements developed for the SSS are of consideration in this paper.

SDIO ROBOTICS PROGRAM FOR SSS

Overview - The robotics contract required that the contractor look at several aspects of the SDS and its followon systems. These included areas within the ground systems where robotics could play a key role in reducing manpower requirements without sacrificing flexibility, capability or security needs. In addition the contractor was directed to look at ground launch operations, on-orbit servicing and manufacturing areas which could gain simi- term and future. This means that the robot be capable of servicing the near term assets of experiments (such as Zenith Star and a proposed Neutral Particle Beam (NPB) experiment in the mid-1990s), the Boost Surveillance and Tracking System (BSTS), the Space Surveillance and Tracking System (SSTS) and the Space Based Interceptor (SBI) as well as those assets anticipated for the future (such as lasers, Space Based Radars (SBR) and NPBs). As one can readily see this is necessarily a very robust environment. Robust in the sense that the robot must be capable of servicing vastly different sizes and styles of spacecraft as well as very different payload requirements and hence their attendent ORU and servicing differences. This robust environment then requires an equally robust robot or else multiple designs for the robotic front end to the SSS. The above translates into the following requirements if a single design is to be able to accomplish all of the above support missions:

1. Volume and weight of ORUs that the manipulator must handle - maximum is 1000 kg and minimum is 1 kg
2. Manipulator arm reach - maximum is twenty meters and minimum is 1 meter
3. Number/types of end effectors/tools - up to five
4. Degrees of freedom - up to seven
5. Human control interface parameters including time lag, tactile and force reflection, and vision feedback are recognized requirements but further study is required before specific quantization may be added
6. Knowledge base to accommodate geometric descriptions of assets to be supported and action to be taken for routine ORU change out and selected contingency manipulations
7. Manipulator hardware and control system capability to provide the desired accuracy and response to constants
8. Provisions for four-color capability including necessary space-to-ground communications bandwidth.

In addition to the above requirements for the manipulator it is recognized that the manipulator design may be simplified if the serviced satellite is designed to be robotically friendly. This means that the satellite must be modular, have a hard-dock capability, have ORUs designed to be robotically removable, be capable of fault detection and isolation to a single ORU 99% of the time and be able to determine a friendly servicer from an unfriendly ASAT vehicle. Most of these requirements are more stringent than the NASA requirements where NASA is working with friendly satellites which may be brought back to the space station or STS for additional support from IVA/EVA astronauts. If NASA requires that the servicer maintain satellites which are not easily brought back to the space station or STS such as the Polar Orbiting Platform (POP) than their requirements will begin to approach SDIOs except for the uniquely military aspects of the SDIO requirements.

One must also be aware that in addition to the above SSS requirements for a robotic servicer that there are additional technology requirements which must be addressed in both the near term and the future in order to insure that there are no "show-stoppers" as one proceeds into the SSS design. These technology issues may be broken down into five areas; systems integration, computer control system, sensors, actuation systems, and man-machine interface. In order to eliminate the possibility of "show-stoppers" the following issues must be addressed in a technology program:

1. Systems Integration Issues
   a. The sensor feedback time lag between ground-based human commands and sensor feedback from the space-based servicer
   b. Architecture definition for the coordination of ground-based and in-space control computers
2. Computer Control System Issues
   a. Suitable uplink to load control programs
   b. Adequate response bandwidths to implement human instructions, and anticipation of manipulator movements where significant time lags between manipulator actions and human controller responses occur
   c. Adequate ground monitoring capability
   d. Communications protocols between SDS space assets and the servicer
   e. Provisions for ensuring safe operation of actuator systems.
3. Sensor Issues
   a. Weight and space adaptation and packaging of available sensor technology into an on-orbit servicer
   b. Adaptation of sensors for satellite stabilization.
4. Actuator System Issues
   a. Development, design and construction of manipulator arms of sufficient dimension
   b. Weight and space adaptation of actuation systems
5. Man-machine Interface Issues
   a. Time lags associated with direct control of the robotic servicer
   b. Input speed due to data throughput speed limitations
   c. Execution precision of commands using direct control
   d. Data saturation due to critical and trivial feedback
6. Unresolved Servicer Issues
   a. What kind, how many and the design required for fluid and electrical connectors for interfacing the servicer to the spacecraft being serviced
   b. What type of docking mechanism shall be used
   c. Final docking sensor developments such as laser ranging, radar, etc.

Each of the above technology issues must be solved as they apply to the SDIO SSS robotic servicer requirements. This means that for an on-orbit servicer to be effective requires that the SDIO initiate parallel efforts in both technology development and SSS development to ensure an effective SSS can be fielded in the mid-to-late 1990's for support of the SDS space assets.

SDS Robotics Program - Due to budget constraints the starting up of an ambitious robotic development program is nebulous. Therefore the SDS Robotics Program must be accomplished with as little cost as possible. This implies some innovative funding must be accomplished. The key is to have the already ongoing NASA/AF/SDIO SSS program carry the bulk of the SDIO requirement satisfaction. In addition there must be some technology programs instituted in order to insure that the SSS program succeeds. These technology requirements can be met by leveraging off of existing programs such as are found within the SDIO's small business and innovative science and technology programs. Also other agencies have on-going robotics efforts which may be useful in achieving the necessary SDIO technology goals with little or no SDIO funding required. These agencies would include DARPA, NASA, and AFSC/WPAL for instance. Efforts are now underway to investigate the feasibility of this approach. Any potential SDIO technology program will probably have to wait until the 1990 budget year regardless.

Applications of SDIO Robotics - A few words about the SDIO applications will help to substantiate the need for robustness of an SSS robotic servicer. The most near term requirement for a SSS in the SDIO program will be to support the mid to late 1990's experiments. The most ambitious of these are the Zenith Star and NPB experiments. These are large vehicles that will likely be too large for a single launch vehicle and thus will require assembly on-orbit. Also the weight limitations may mean that fuel and other depletables will be minimized in order to ensure that the experiment is a success. It would appear that if these satellites were to have provisions for on-orbit servicing that not only could their on-orbit lifetime be extended but also provision for ORU changeout may provide for a way of upgrading or changing the satellites design should that be necessary to ensure success. The next application would be to the SDS Phase I implementation in the mid to late 1990's. This would include the BSTS, SSTS and SBI satellite programs. Several studies such as the SAMS study have shown that the capability to be able to service satellites on-orbit would provide for life-cycle cost savings. This savings comes from being able to keep the satellites alive after replenishable depletions, to provide a way of repairing a failed satellite without launching a new satellite, to provide for upgrading the satellite with new ORUs without having to design, develop and launch a new constellation and other possibilities which will become obvious once designers realize that their satellites are accessible after launch. And finally application to the followon SDS assets which will have another set of requirements associated with them such as on-orbit assembly, alignment, and calibration requirements that do not exist on the more near term satellites.

Conclusion - The SDIO has been investigating the feasibility of on-orbit servicing through a succession of studies and design and development efforts. Paramount to these efforts achieving their goals is that a proper robotic front-end be available to the SSS. The SDIO's robotic study has developed the necessary requirements that this robotic servicer must meet in order to achieve these on-orbit servicing goals. Only through judicious and innovative application of scarce funds can we hope to have a successful robotics program and associated SSS program.

CREW CHIEF ANIMATION TECHNIQUES: DATA APPLICATIONS FROM A SPACE SUIT STUDY

M. Korna and Captain A. Reinhardt
Wright-Patterson AFB

(Paper not provided by publication date.)

CREW CHIEF: A COMPUTER GRAPHICS SIMULATION OF AN AIRCRAFT MAINTENANCE TECHNICIAN

Nilss M. Aume
Harry G. Armstrong Aerospace Medical Research Laboratory
Wright-Patterson AFB, Ohio 45433

ABSTRACT

Approximately 35% of the lifetime cost of a military system is spent for maintenance. Excessive repair time is caused by not considering maintenance during design. Problems are usually discovered only after a mock-up has been constructed, when it is too late to make changes. CREW CHIEF will reduce the incidence of such problems by catching design defects in the early design stages. CREW CHIEF is a computer graphic human factors evaluation system interfaced to commercial CAD systems. It creates a three dimensional man model, either male or female, large or small, with various types of clothing and in several postures. It can perform analyses for physical accessibility, strength capability with tools, visual access, and strength capability for manual materials handling. The designer would produce a drawing on his CAD system and introduce CREW CHIEF in it. CREW CHIEF's analyses would then indicate places where problems could be foreseen and corrected before the design is frozen.

INTRODUCTION

CREW CHIEF is a computer graphics model of a maintenance technician. This model, interfaced to existing commercial CAD systems used by aerospace manufacturers, may assist in evaluating the maintainability of Air Force aircraft as well as equipment in general. CREW CHIEF is an expert system which enables the designer to perform the functions of an expert ergonomist. The CREW CHIEF model allows the designer to simulate a maintenance activity using the computer-generated design as a 3-D mockup to determine if the required maintenance activities are feasible. CREW CHIEF is a large program, with a core of more than 500 subroutines containing over 120,000 lines of code. Version 1 of the CREW CHIEF model was completed in April 1988, and has been distributed to major DoD contractors.

Since CREW CHIEF is interfaced to existing commercial CAD systems used by aerospace manufacturers, the program does not require users to transfer the system or equipment design into the CREW CHIEF program for analysis; rather, CREW CHIEF is called into the user's drawing without any conversion.

The CREW CHIEF model provides for the designer the ability to simulate, on the computer-aided drawing board, maintenance and other related human operator interactions with a system. It creates a model of the correct body size and proportions of both male and female maintenance technicians, the encumbrance of clothing and personal protective equipment (PPE), mobility limitations for simulating working postures, physical access for reaching into confined areas (with hands, tools, and objects), visual access (evaluating what the CREW CHIEF can see), and strength capability (for using hand tools and manual materials handling tasks).

The CREW CHIEF user can select from a range of body sizes of male and female technicians. CREW CHIEF will generate 1st, 5th, 50th, 95th, and 99th percentile body sizes for male and female maintenance technicians. During a CREW CHIEF run, at selected points in the program, the user can select other genders and body sizes to be used in further analyses.

CREW CHIEF automatically considers the encumbrance of clothing and PPE, a very important limitation for the maintenance technician. The initial model has four types of standard clothing to choose from: fatigues, cold weather, arctic, and chemical defense. The clothing types add encumbrance to the joint mobility limits when the human-model performs reaches. The program

adjusts the posture to one used in maintenance activities, and analyzes the accessibility considering the clothing type selected.

The model automatically plots the workplace from the CREW CHIEF human-model's viewpoint to evaluate visual accessibility, showing appropriate obscuration of gear worn on the head. The visibility plot allows the designer to see the task from the maintenance technician's viewpoint to verify visual accessibility. Both a CRT presentation and a hard copy plot are available.

To simulate the body postures typical in maintenance, CREW CHIEF provides twelve starting postures: standing, sitting, kneeling on one knee, kneeling on both knees, bending, squatting, lying prone, lying supine, lying on the side, walking, crawling, and climbing. Some of these postures reduce the mobility and strength available to perform the task. The designer should initially select the most appropriate posture, and then he/she can manipulate all the body segments to change a posture to make it fit the situation. Automated evaluations of accessibility, reach, and strength analyses can be performed in each of these postures.

The CREW CHIEF program computes the strength capabilities of the maintenance technician based on gender, posture, and the task performed. CREW CHIEF computes strength for manual materials handling tasks (lifting, carrying, holding, pushing, and pulling), applying torque to bolts using wrenches, and connecting /disconnecting electrical connectors. More than 100,000 strength measurements were conducted to develop the strength analysis models in CREW CHIEF.

Accessibility analysis capabilities include the ability of the human-model to reach and operate any tool or object. The object, an electronics box for example, may already be part of the design. The CREW CHIEF program has 105 common hand tools to evaluate reach and accessibility. These were modeled from the tools available in Air Force maintenance technicians' tool boxes. By providing the designer a set of common tools for evaluation, the need for special tools will be reduced. The designer chooses the tool to be used and, depending on the tool, there may be one or several methods of holding the tool in the hand.

The CREW CHIEF model is three-dimensional. To accurately represent clothing, the model has a surface of thousands of triangular facets attached to the 35 links of the skeletal link system. The designer may select the simplified 3-D model when rotating the workplace on the display and a higher resolution 2-D profile projection for plots. The user may also produce on-screen shaded surface pictures from this model.

The largest single effort in the development of the CREW CHIEF human-model was gathering the research data. The CREW CHIEF human-model is a simulation of the physical characteristics and limitations of the maintenance technician. The development of this simulation requires an extensive and accurate data base describing those characteristics and limitations.

CREW CHIEF has been provided to government contractors. Later, it will be available to other industries. The software is available without charge, but users must have the appropriate computer hardware and software. At present, CREW CHIEF Version 2 with a common core has been interfaced to the following host CAD systems:

- CREW CHIEF Host-Independent (unhosted core of CREW CHIEF) FORTRAN 66 and FORTRAN 77.

- CREW CHIEF-CADAM Version 20 with Geometry Interface Module for MVS/SP operating system, FORTRAN 66H.

- CREW CHIEF-CADAM Version 21 with Interface User Exit (IUE) for MVS/SP operating system, VS FORTRAN 66 and 77, and for VM/IS operating system, FORTRAN 77H Extended.

- CREW CHIEF-Computervision Version CADS 4001 with an Analytical Processing Unit (APU) and CADDS 4X software, revision 5B or later.

- CREW CHIEF-Computervision CADDStation Version for UNIX operating system.

In addition to the previously described access, strength, and visibility analyses, work is currently underway on two more capabilities for CREW CHIEF. First, a Task Time Estimator will be added. While a large body of knowledge exists concerning performance times on various actions, a well known example being MTM or Methods Time Measurement, these typically are actions performed under fairly benign conditions. In aircraft maintenance, the technician frequently has to work under rather severe constraints. e.g., in a prone position, in hard to reach and invisible locations, in heat and cold, and while wearing protective clothing, to name a

few. Conditions like these prolong performance times. When completed, the Task Time Estimator will be able not only to predict the basic performance time but also by how much the time will be prolonged because of the various detriments discovered during the analyses.

The other future undertaking is to model an astronaut, wearing a space suit, and working under microgravity conditions. Two things are immediately obvious: one, strength data from one G will be largely invalid, and two, the mobility of the astronaut is very much determined by the design of the space suit. To gather strength data, an initial experiment has been conducted under water to simulate microgravity. Still large amounts of work remain to be done before the data are of a sufficient quantity to be usable. While much of the work will be done under water, it may possible to gather some under actual orbital conditions.

Modeling the astronaut's mobility will require studying and measuring the design of the space suit. The directions and magnitudes in which a space suit can move will be the limiting factors in this case. Also, the strength needed to move the space suit will have at least some effect on the available strength of the astronaut. There may be postures, possible under microgravity, that are are impossible to attain on earth. While the final decision on this has not been made, it is conceivable that the space program will not be part of CREW CHIEF but will be an independent and separate program. Otherwise, the result may be a program that is just too unmanageably large.

REFERENCES

Korna, Medhat, et al, "User's Guide for CREW CHIEF: a Computer Graphics Simulation of an Aircraft Maintenance Technician," AAMRL-TR-88-034, Armstrong Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, May 1988.

Modeling Strength Data for CREW CHIEF

Joe W. McDaniel, Ph.D.
Armstrong Aerospace Medical Research Laboratory
Wright-Patterson AFB, Ohio 45433

ABSTRACT

The Air Force has developed CREW CHIEF, a computer-aided design (CAD) tool for simulating and evaluating aircraft maintenance to determine if the required activities are feasible. CREW CHIEF gives the designer the ability to simulate maintenance activities with respect to reach, accessibility, strength, hand tool operation, and materials handling.

While developing the CREW CHIEF, extensive research was performed to describe workers strength capabilities for using hand tools and manual handling of objects. More than 100,000 strength measures were collected and modeled for CREW CHIEF. These measures involved both male and female subjects in the 12 maintenance postures included in CREW CHIEF. This presentation describes the data collection and modeling effort.

INTRODUCTION

Early identification of potential design-induced maintainability problems is essential to correct a problem before mock-up, fabrication, or production. To facilitate early identification of design problems, the Harry G. Armstrong Aerospace Medical Research Laboratory (AAMRL) and the Air Force Human Resources Laboratory (AFHRL) have developed CREW CHIEF, a computer-aided design (CAD) model of an aircraft maintenance technician.

Approximately 35 percent of the lifetime equipment cost and one-third of all manpower is spent on maintenance. Excessive repair time is caused by failure to adequately consider maintenance demands. The maintenance technician will spend hours making a repair which could have been completed in minutes with better accessibility. The CREW CHIEF model will reduce the incidence of such problems by allowing the designer to perform maintainability analyses and correct design-related defects. Ultimately, not only will development engineering costs and acquisition time be reduced, but also life cycle costs and maintenance time while system availability grows.

Accessibility is a major problem in maintenance. Objects being maintained do not usually have the faulty components located for the convenience of the maintenance technician. Anything can fail, and eventually does if its used long enough. So virtually every detail of every component is a candidate for maintenance. Equipment designers attempt to place the high failure rate items in more accessible locations, but the function of the component usually takes precedence in determining location. Also, when new equipment is being designed, the failure rates for components are only estimates, and these estimates sometimes turn out to be far from accurate. The result is that high failure rate components are sometimes in inaccessible locations.

This results in "work arounds", where the maintenance technician is forced to work in uncomfortable and inefficient postures, such as kneeling, bending, squatting, prone, supine, lying on the side, or sitting on the ground. These are the "everyday" working postures for maintenance technicians. Because these postures are uncomfortable and and less stable than standing or sitting in a chair, we can predict that they are less efficient. We can readily observe that the time required to perform a task in these "maintenance" postures is longer than the traditional postures of sitting or standing. We also know that the forces generated by the worker's strength will be less because of the less common directions of force and the less stable support for the body.

The body's ability to generate force varies greatly with the direction of force. A combination of gravity and body posture is the cause of this phenomenon. Most of our exertions tend to be performed with the trunk in a more or less erect posture and applying a force to overcome gravity, as in lifting and lowering of objects. The next most frequent activity involves pushing and pulling, that is, exerting a force away from or toward the body. Because we are mobile, we tend to avoid lateral force exertions. It is usually possible to "face" the work, so that lateral forces are minimized. These activities constitute most of what we define as manual materials handling. Tests confirm that the muscles produce relatively more force in these directions than in other less used postures and directions.

THE CREW CHIEF MODEL

CREW CHIEF, a computer-aided design (CAD) model of an aircraft maintenance technician which allows the designer to perform the functions of an expert ergonomist. The designer may simulate a maintenance activity on the computer generated image to determine if the activity is feasible. Expert system software automatically creates the correct body size and proportions for males and females, the encumbrance of clothing, personal protective equipment, and mobility. Physical access for reaching into confined areas (with hands, tools, and objects), visual access, and strength.

Version 1 of CREW CHIEF was completed in April, 1988. It incorporates several data bases, functional capability in 12 different maintenance postures, 1st to 99th percentile male and female dimensions, and 4 clothing types. Task analyses include flightline tools and manipulating components. Visibility and task interference analyses can be computed with this "electronic mock-up". More than 30 copies of CREW CHIEF software have already been released to aerospace companies. CREW CHIEF has already been used to support Air Force and Army programs. Version 2, almost complete, features an enhanced tool data base, tool envelop analyses, 3-D shaded surface enfleshment for the man-model, alternate populations, and animation.

To simulate the body postures typical in maintenance, CREW CHIEF provides twelve starting postures: standing, sitting, kneeling on one knee, kneeling on both knees, bending, squatting, lying prone, lying supine, lying on the side, walking, crawling, and climbing. Some of these postures affect the mobility and strength available to perform the task. Appropriate evaluations of accessibility, reach, and strength analyses can be performed in each of these postures.

The CREW CHIEF program computes the strength capabilities of the maintenance technician based on gender, posture, and the task performed. CREW CHIEF computes strength for manual materials handling tasks (lifting, carrying, holding, pushing, and pulling), applying torque to bolts using wrenches, and connecting /disconnecting electrical connectors. More than 100,000 strength measurements were conducted to develop the strength analysis models in CREW CHIEF.

Accessibility analysis capabilities include the ability of the human-model to reach and operate any tool or object. The object, an electronics box for example, may already be part of the design. The CREW CHIEF program has 105 common hand tools to evaluate reach and accessibility.

STRENGTH RESEARCH

The largest single effort in the development of the CREW CHIEF human-model was gathering the research data. The CREW CHIEF human-model is a simulation of the physical characteristics and limitations of the maintenance technician. The development of this simulation requires an extensive and accurate data base describing those characteristics and limitations.

Table 1 shows combinations of variables and types of strength measured for the CREW CHIEF model development. More than 100,000 strength measures were made. An "X" indicates that a particular variable and type of strength variable was researched. In most cases the "X" represents a number of individual studies. For example, for the first combination of standing and tool torque, seven separate studies were performed with different combinations of other variables. Other combinations of variables included different sizes and lengths of wrench handles, different orientations of wrench handles and bolt heads, loosening and tightening exertions, different combinations of hands (right, left, or both), different types and sizes of wrenches, with and without gloves, different types and degrees of obstructing barriers, and extensions and U-Joint sockets.

A Sonic digitizer was used to measure body posture in many of the strength studies. Posture is an impor-

VARIABLES TESTED FOR CREW CHIEF STUDY

VARIABLE	TOOL TORQUE	LIFT	PUSH & PULL	CARRY	HOLD & POSITION	CONNECTOR
GENDER	X	X	X	X	X	X
OBJECT HEIGHT	X	X	X		X	X
ORIENTATION	X	X	X		X	X
BARRIERS	X			X	X	X
HANDLE SIZE	X					X
ONE HANDED	X	X	X	X	X	X
TWO HANDED	X	X	X	X		
POSTURE						
STAND	X	X	X		X	X
SIT	X	X	X		X	
BEND	X	X	X	X	X	X
SUPINE	X	X	X			
PRONE	X	X	X			
SIDE	X	X	X		X	
KNEEL	X	X	X		X	
SQUAT	X	X	X	X	X	
WALK				X		
CRAWL			X	X		

TABLE 1. Combinations of variables and types of strength measured for the CREW CHIEF model development. More than 100,000 strength measures were made. An "X" indicates that a particular variable and type of strength variable was researched. The "X" usually indicates a number of individual studies were performed on a particular combination, with additional combinations of variables not shown in this table.

tant variable when shifting the center of body mass effects the force generated. For example, in pushing or pulling, the body mass may be shifted by bending or straightening the elbows. In one study of pushing strength, for example, men averaged 48 percent more and women 30 percent more when pushing with bent elbows versus straight elbows.

The sonic digitizer employs an array of microphones surrounding the subject. Electric spark gaps are taped to the subject's joint centers or other anatomical features useful in tracking posture. The sonic digitizer measures the time delay between the generation of the spark and when each of the microphones detect the popping sound of the spark. The delay is translated into a slant range distance, then the 3D coordinates are computed. The sonic digitizer can locate points in 3D space at the rate of 40 Hz. By surrounding the subject with an array of 8 microphones, masking of body parts is eliminated.

Posture is critically important in all types of physical activities. It is especially important in maintenance tasks where the object being maintained often creates obstacles and forces the worker into restricted postures. When carrying for example, a low ceiling in a passage way or under the wing or fuselage of an aircraft can reduce the available strength, as shown in Table 2 below. The low ceiling (40% of stature) forces the worker into a bent posture, then a semi-squat, and finally crawling. At each progressive level, the amount of

weight that an individual can carry is reduced, until, in a crawling posture, it averages only 45 percent of the no restriction condition.

CEILING HEIGHT	MALE	FEMALE
UNLIMITED	153	79
80%	146	73
60%	113	54
40%	64	41

TABLE 2. Maximum weight (pounds) that can be carried in an 18 inch wide box with no handles while using two hands. The ceiling height was set to a percentage of stature. Values are averages.

The effect of posture on weight lift capability can be seen in Table 3. The values shown are the maximum weight (in boxes) people were able to lift from the floor and place on a shelf. The subjects were 50 men and 50 women. The maximum weight was determined using the incremental technique, increasing the amount of weight lifted until the subjects were no longer able to lift the box onto the shelf. Five postures were used: standing, kneeling, sitting, squatting, and lying on the side. All lifts used two hands, and the height of the shelf was adjusted to 35 percent of the subject's reach height in that posture, except for lying, where the shelf was ten inches high. The values are in pounds and averaged over each group of subjects.

POSTURE	MALES	FEMALES
Standing	118	58
Kneeling	99	53
Sitting	92	49
Squatting	79	43
Lying, side	42	21

TABLE 3. Maximum Weight Lift (Pounds) Capability for Different Postures. Maximum weight that can be lifted and placed on a shelf at chest height in a 24 inch wide box with no handles while using two hands.

The table shows the effective strength decreases when the body support becomes less stable. The kneeling posture allows more mobility of the lower torso in adjusting the posture toward the load while still providing a stable support. Sitting provides a stable support but reduces the mobility of the lower torso, forcing the reach to shelf to be farther. The squatting posture has little support, as the subject is supported by the balls of the feet and must exert some effort to maintain balance. The lying on the side posture has little stability in the axis of load and requires the exertion of lateral forces to raise the weight.

A maintenance technician would prefer to work standing up. Often this is not possible because of constraints in the workplace. Obstructions limit the access, forcing a less than desirable posture.

Another example of the interaction of posture and direction of force is found in measures of torque produced with a socket wrench. In a study of isometric torque measured on 20 men and 20 women using a 1/2-inch square drive ratchet to turn a bolt with a 3/4 inch head, it was found that the least favorable location/orientation of a bolt head allowed only fifteen percent of the torque produced in the most favorable location/orientation.

Of course a maintenance technician would not choose the less favorable configuration, but may be forced to do so by obstacles. In another study measuring wrench torque, but where the subject had to reach over or around obstacles in the workplace, the available torque was reduced up to 80 percent due to the obstacles.

Ergonomics data is limited in supply and not familiar to most equipment designer. Designers typically think of the more ideal circumstances when considering the maintainability of a design. Furthermore, designers tend to overestimate the strength capabilities of the maintenance technician, especially failing to discount the strength do to awkward postures. If a significant portion of maintenance technicians are less strong than the designer imagines, impossible tasks may be inadvertently created.

MODELING STRENGTH

Many previous ergonomics models have failed to achieve their goal simply because model developers incorrectly assumed that all required data was available. There is a vast quantity of data available in the ergonomics literature, but most are not suitable for

development of a general purpose model. Most data are limited in the range of variables, the sample size, the applicability of the subjects to military populations, and non-availability of raw data for modeling. Developers of the Crew Chief model have programmed a large portion of their resources for ergonomics research. CREW CHIEF developers have gathered data regarding manual materials handling for the appropriate working postures, and torque strength capability for wrenches and electrical connectors. After the model development, these data will be submitted for inclusion in traditional military standards.

To overcome the limitations described above, a seven step testing and modeling procedure was developed for the CREW CHIEF program. Because of the complexity and the amount of research data needed for the CREW CHIEF model, it was not possible to gather all data on a representative sample of maintenance personnel. Rather, a benchmarking technique was developed to allow laboratory research to represent the population of workers. This seven step process insures that research data is representative of the population of Air Force maintenance personnel:

First, subjects were screened to represents the size and age of Air Force maintenance personnel. Since more than 99 percent of personnel doing manual work are age 30 or younger, research subjects were limited to the range of 18 to 30 years. The Air Force also has strict height and weight allowances defined by Air Force Regulation 160-43 which were also applied to research subjects. These restrictions may limit the utility of CREW CHIEF to represent older civilian populations, but Army and Navy personnel have almost identical characteristics.

Second, subjects were given benchmark strength tests. This battery of tests has been given to large samples of Air Force maintenance personnel over the years. One of these is the Maximum Incremental Weight Lift to Six Feet. This test is given to all Air Force and Army recruits and has been demonstrated to be highly correlated with manual materials handling tasks. Three static (isometric) strength tests are also given: the one-arm pull, which involves bracing the straightened left arm while pulling on a vertical handle with the right; the elbow height lift, which involves lifting against vertical handles positioned at elbow height; and the 38cm lift, which involves lifting with two hands against a horizontal handle 38 cm above the floor. These tests have also been given to several thousand military personnel.

Third, the subject's body size is measured. For subjects tested at AAMRL, 69 measures were taken, for some of the tests made off-site, 20 measures were taken. These measures were made on several thousand military personnel.

Fourth, the subjects participated in simulated working tasks wherein their strength was measured. In most of these simulated work tasks, from 40 to 100 subjects were tested in each combination of variables. Treatments were randomized with suitable rest periods between all strength measures. Some treatment conditions were repeated at both the beginning and the end of each test session to verify the reliability of the subject's performance. The tool torque and push/pull tests were static (isometric) while the lifting, carrying, and holding were dynamic. Static measures were gathered with a computerized data collection system which evaluated the data against goodness criteria as it was collected and identified exertions to be repeated.

Fifth, the data were sorted, collated and edited. This process used both within subject and between subject relationships to identify outlying data values.

Sixth, the data were adjusted to represent the population of workers. This was accomplished using regression equations developed on large samples performing both the benchmark tests and some of the work tasks.

Seventh, the adjusted data were converted to algorithmic models for CREW CHIEF. When user of CREW CHIEF defines a task to be performed, the model determines which conditions apply, and select the appropriate strength models of male or female data. Predicted strengths for the 1st, 5th, 50th, 95th, and 99th percentiles are displayed on the workstation.

CONCLUSION

Most published ergonomics data are not suitable for development of a general purpose model because of limitations in the range of variables, the sample size, the applicability of the subjects to military populations, and non-availability of raw data for modeling. Developers of the Crew Chief model have developed an integrated procedure for defining data needs in terms of the tasks to be modeled,

selecting representative subjects, developing benchmarking techniques for matching laboratory data to population characteristics, analyzing and modeling the data, presenting data to designers in a comprehensive computer-aided tool.

CREW CHIEF developers have gathered data regarding manual materials handling for the appropriate working postures, and torque strength capability for wrenches and electrical connectors. These data have verified assumptions that posture and accessibility greatly limit the forces that can be generated by human strength and that posture and accessibility must be explicitly considered in the design of the maintenance workplace.

ACKNOWLEDGMENTS

The author wishes to acknowledge the contribution of other Air Force colleagues Mr. Nilss M. Aume, also of AAMRL and Ms. Jill Easterly of the Air Force Human Resources Laboratory as co-developers of the CREW CHIEF program.

The University of Dayton Research Institute has had 20 to 25 full-time professionals headed by Mr. Medhat Korna working on the CREW CHIEF development for the last five years.

Dr. M. M. Ayoub of Texas Tech University and his staff was largely responsible for all the collection of weight lifting and holding data.

Dr. S. Deivanayagam of Tennessee Tech University (formerly with University of Texas at Arlington) was responsible for two of the studies involving tool torque.

REFERENCES

1. McDaniel, J. W., "Tools for Ergonomic Accommodation: COMBIMAN and CREW CHIEF", COMPUTER-AIDED DESIGN: APPLICATIONS IN ERGONOMICS AND SAFETY, Ed. W. Karwowski, Taylor and Francis, London, 1989.

2. Korna, M., Krauskopf, P., Haddox, D., Hardyal, S., Jones, M., Polzinetti, J., and McDaniel, J., "User's Guide for CREW CHIEF: A computer Graphics Simulation of an Aircraft Maintenance Technician (Version 1-CD20)", AAMRL-TR-88-034, Armstrong Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio, July, 1988.

THE USE OF PLAID IN THE SPACE STATION FREEDOM VIEWING ANALYSIS

Frances E. Mount
NASA/JSC
Houston, Texas 77058

Sandra D. McKee
Lockheed Engineering & Sciences Co.
Houston, Texas 77058

INTRODUCTION

The focus early in the Space Station Freedom Program was on identifying viewing requirements for both direct viewing (windows) and indirect viewing (closed-circuit television). These requirements now reside in NASA 's Program Definition and Requirements Document (PDRD), Section 3: Space Station Systems Requirements.

Currently, an analysis is being conducted to address the feasibility of viewing for Space Station Freedom. The goal of this analysis is to determine the optimum location for the windows and closed-circuit television cameras in order to meet the established requirements. Additionally all viewing provisions must adequately support Space Station assembly and on-board operations once the Station is operable.

PLAID, a three-dimensional graphics system developed at NASA/Johnson Space Center, was selected for use as the primary tool in this analysis. PLAID provides the capability to simulate the assembly of the Space Station Freedom, as well as to examine operations on the Station as it evolves. In addition, it is used as a tool to analyze general out-the-window viewing conditions for all Space Station components, and provides the ability to integrate an anthropometric scale-modeled person (representing a crewmember) with the Station's interior architecture.

PROCESS

Candidate window and camera locations were selected, and relevant viewing requirements were identified. Using the PLAID system, a geometric model of each Space Station Freedom element and the overall configuration were created reflecting the current Station architecture, both internal and external. Proposed window locations were included in these models.

Anthropometric models of crewmembers, provided by PLAID, were used in a simulation of out-the-window viewing. Pictured in Figure 1 is a 95th-percentile male crewmember. This system also enabled the users to analyze the interface occurring between the crewmember and the Space Station structure itself. Represented in Figure 2 are two 50th-percentile male crewmembers involved in a Space Station cupola viewing task. Figure 3 depicts a 50th-percentile male crewmember in the wardroom area.

Using these tools, the analysts were able to identify precisely how well each requirement is satisfied by a proposed window location. Optimum window placements were determined by integrating the view available from a particular position with various other factors such as operations requirements, off-duty and on-duty activity viewing needs, anthropometric clearance, optimum traffic flow, and window accessibility.

CONCLUSION

The PLAID system, through graphic analysis, has provided a means to optimize window and camera placement on Space Station Freedom and to determine if viewing requirements have been met.

The system has the capability to accurately depict:

(1) the window and/or camera view as seen through a specified eyepoint at each candidate location.

(2) the clearance limitations imposed by the Space Station architecture and anthropometry of crewmembers.

Combining these two factors, window and camera locations have been selected in order that viewing stations will provide adequate views which meet the specified requirements and will be comfortable to use.

ACKNOWLEDGEMENTS

This viewing analysis could not have been accomplished without the NASA Graphics Analysis Facility and the Lockheed Software Systems Section.

REFERENCES

1. Mount, Frances E. and Sandra D. McKee, "Space Station Freedom Viewing Analysis": Volumes 2, 3, 4, and 7, JSC No. 32089, NASA/JSC, Houston, Texas, 1989.

2. Mount, Frances E. and James L. Lewis, "Space Station Viewing Requirements", 1986 SAE Aerospace Technology Conference and Exposition, Long Beach, California, 1986.

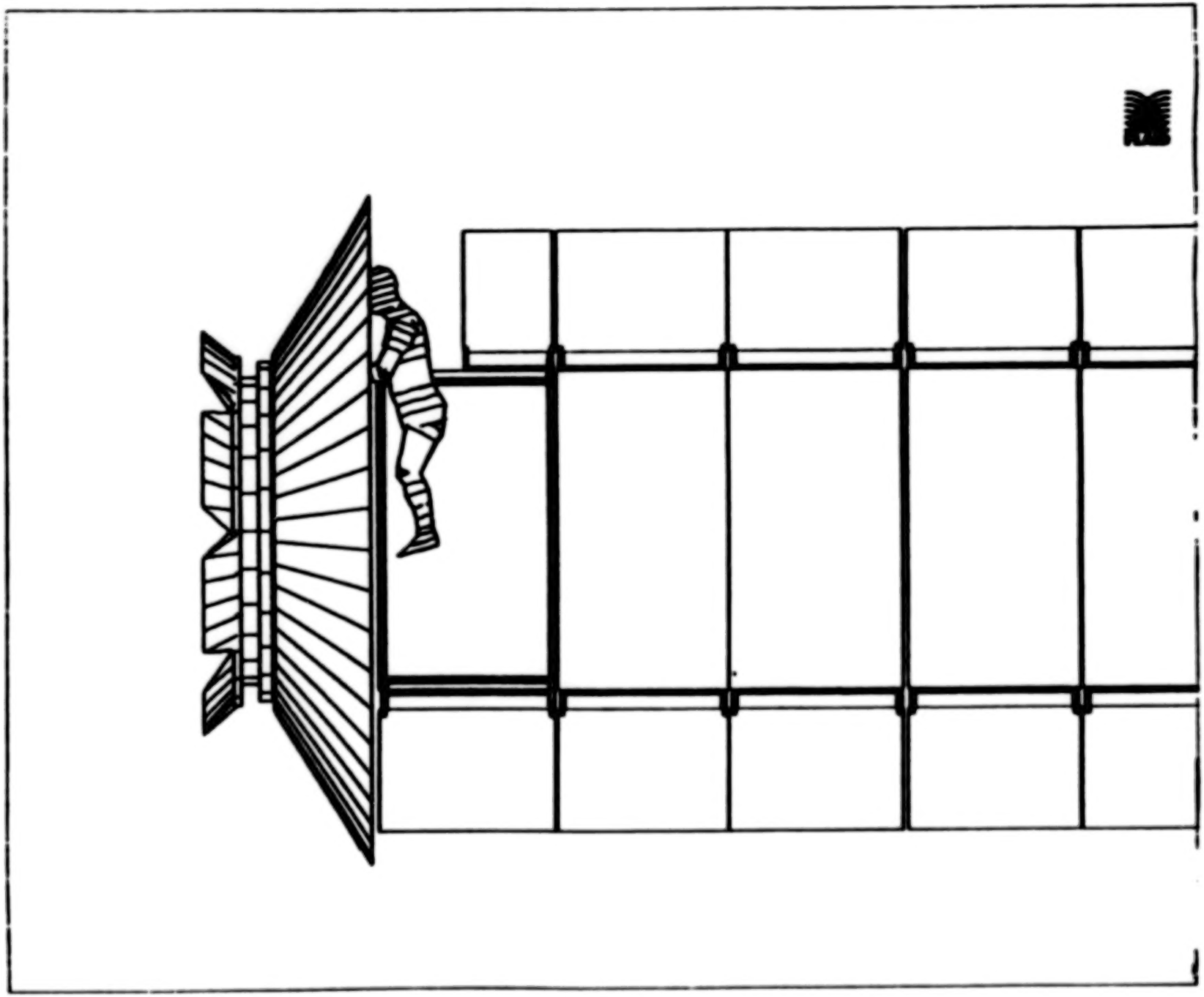


Figure 1

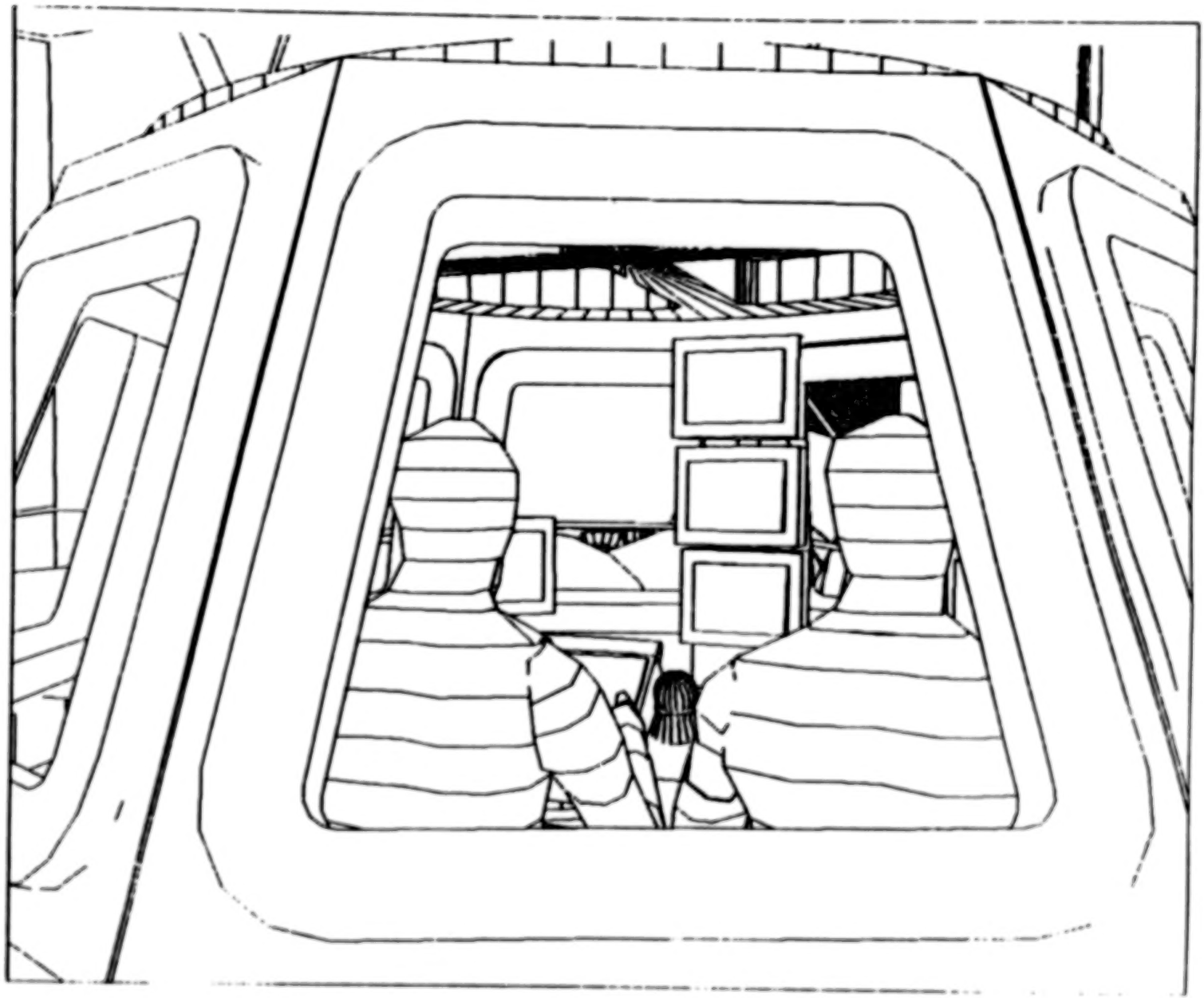


Figure 2

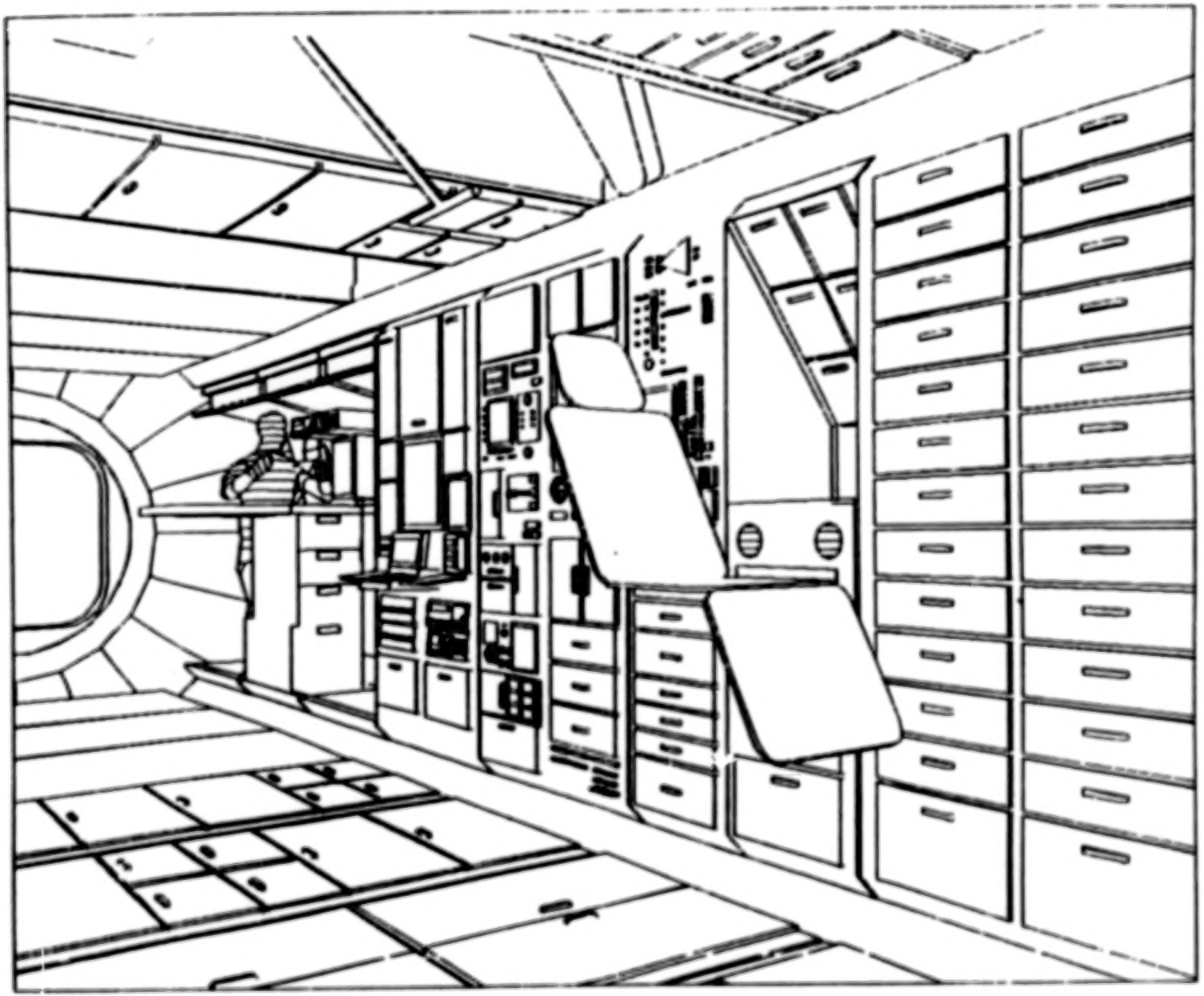


Figure 3

153

QUANTITATIVE ASSESSMENT OF HUMAN MOTION USING VIDEO MOTION ANALYSIS

John D. Probe
Lockheed Engineering and Sciences Company
2400 Nasa Road 1 / C95
Houston, Texas 77058

INTRODUCTION:

In the study of the dynamics and kinematics of the human body a wide variety of technologies has been developed. Photogrammetric techniques are well documented and are known to provide reliable positional data from recorded images. Often these techniques are used in conjunction with cinematography and videography for analysis of planar motion, and to a lesser degree three-dimensional motion. Cinematography has been the most widely used medium for movement analysis. Excessive operating costs and the lag time required for film development coupled with recent advances in video technology have allowed video based motion analysis systems to emerge as a cost effective method of collecting and analyzing human movement. The Anthropometric and Biomechanics Lab at Johnson Space Center utilizes the video based Ariel Performance Analysis System to develop data on shirt-sleeved and space-suited human performance in order to plan efficient on orbit intravehicular and extravehicular activities.

Video Based Motion Analysis:

The Ariel Performance Analysis System (APAS) is a fully integrated system of hardware and software for biomechanics and the analysis of human performance and generalized motion measurement. Major components of the complete system include: the video system, the AT compatible computer, and the proprietary software.

Video Subsystem:

The video system consists of commercial quality 1/2" VHS format portable video cameras. They are used to record motion sequences for subsequent analysis. A minimum of two cameras are required for full three-dimensional analysis. A high quality VCR and monitor are used for the display and digitizing of videotaped sequences. The playback unit accommodates standard VHS videotapes recorded from any standard video source to allow high precision freeze-frame video imaging with accurate single frame advance and reverse as well as a variable speed search capability.

Hardware Subsystem:

An AT compatible computer is the primary component of the analysis system. The computer uses a combination of a frame grabber and a VCR controller board to digitize from the playback unit. The APAS captures video images from video tape and imports them into the computer memory. The operator can then digitize the desired sequence by positioning a cross-hair cursor over the joint center of interest and record the coordinates of this point by pressing a button on the mouse. Digitization of the joints on the first frame is performed completely by the operator. For subsequent frames, the point locations from previous frames are used to predict the positions of each point on the current frame. This significantly reduces the time required to

digitize a sequence. Additionally, since the computer stores a digital image, the analysis frame can be contrasted, enhanced, or filtered for clarification.

Software Subsystem:

An extensive integrated software package makes up the third component of the APAS. For ease of operation, the software has been highly structured and modularized. Each module is designed to perform a particular function and is completely menu driven. A brief functional description of each module is listed below.

Performance analysis always begins with the DIGITIZING module. This module allows video images to be converted to body joint location coordinates in the computer. These digitized locations are saved for subsequent conversion to true image space location.

The TRANSFORMATION module converts digitized video data into true two or three-dimensional image data using an algorithm called direct linear transformation (1). If a single camera view is used, the resulting image is two-dimensional. If two or more camera views have been used, the resulting image is three-dimensional.

The SMOOTHING module removes small random digitizing errors from the computed image coordinates. At the same time, it computes body joint velocities and accelerations from the smoothed joint coordinates. The operator may choose any of three different smoothing functions: cubic spline, digital filter, and polynomial smoothing. The APAS utilizes, as the smoothing method of choice, a modified cubic spline smoothing algorithm formulated by Reinsch (2). In addition, the user may control the amount of smoothing applied to each joint to insure that smoothing does not distort the digitized data.

The VIEWING module is used to examine image data in "stick figure" format. Viewing options include single frame, multiple frame, and animated images. Three-dimensional images may be rotated to allow viewing from any chosen direction.

The GRAPHING module is used to draw graphs of image motion. Displacement, velocity, and acceleration curves may be graphed for any number of individual body joints or segments. Joint motion may be presented in either linear coordinates or angular coordinates, while segment motion is presented in angular coordinates about a single segment endpoint. The data may be displayed on cartesian graphs or as full figure models.

Printed reports of the image motion data are produced by the PRINT module. Data can be saved for future printing. Additionally, reports may be transferred to other systems such as spread sheets or data base programs.

The ANALOG module includes a hardware interface and an analog sampling unit with program selectable gain for collection of up to 16 channels of analog input. Specialized features support the use of force plates and electromyography (EMG) measurement and analysis. Included are options for spike analysis, envelope processing, signal integration analysis, waveform analysis, and spectral analysis.

Studies In Progress:

The Anthropometric and Biomechanics Lab (ABL) is currently involved in ongoing studies to enhance astronaut performance in a space environment. Depending on the study, all or part of the APAS may be used for data collection and analysis.

Initial investigations are in progress utilizing motion analysis, EMG, and an instrumented treadmill to measure and compare the shirt sleeved one-gravity relations between velocity, angle of inclination, skeletal muscle contraction patterns, and impact loading of the skeletal system to identical conditions in a zero-gravity environment.

An unrelated but similar investigation is in progress to determine the lower torso

mobility of one of the space station prototype space suits in 1-gravity, in addition to simulated lunar and martian gravity, 1/6 and 1/3 earth gravity respectively. The treadmill and EMG are also being incorporated into this study.

The ABL is also investigating the possibility of using the APAS to determine reach envelopes of astronauts as a function of varying gravitational loads while wearing the Launch/Entry Suit (LES).

CONCLUSION:

The video based motion analysis system being used by the ABL has proved to be a viable means for collecting and analyzing human motion. A great strength of video based systems is their flexibility. The system is used in the one-gravity lab environment, in neutral buoyancy at JSC's Weightless Environment Training Facility (WETF) , and in zero-gravity onboard NASA's KC-135. The systems are versatile and allow the operator to analyze virtually any motion that can be sequentially imaged.

REFERENCES:

1. Shapiro, R. "Direct Linear Transformation Method For Three-Dimensional Cinematography." RESEARCH QUARTERLY, 49:197-205, 1978.

2. Reinsch, C. H. "Smoothing By Spline Functions." NUMERISCHE MATHEMATIK, 10:177-184, 1967.

GASEOUS OPTICAL CONTAMINATION OF THE SPACECRAFT ENVIRONMENT: A REVIEW

N. H. Tran, M. A. Maris and I. L. Kofsky
PhotoMetrics, Inc., 4 Arrow Dr., Woburn, MA, 01801

and E. Murad
Air Force Geophysics Laboratory, Hanscom AFB, MA, 01731

ABSTRACT

Interactions between the ambient atmosphere and orbiting spacecraft, sounding rockets, and suborbital vehicles, and with their effluents, give rise to optical (extreme UV to LWIR) foreground radiation which constitutes noise that raises the detection threshold for terrestrial and celestial radiations, as well as military targets. We review the current information on the on-orbit optical contamination. Its source species are created in interaction processes that can be grouped into three categories: 1) Reactions in the gas phase between the ambient atmosphere and desorbates and exhaust; 2) Reactions catalyzed by exposed ram surfaces, which occur spontaneously even in the absence of active material releases from the vehicles; and 3) Erosive excitative reactions with exposed bulk (organic) materials, which have recently been identified in the laboratory though not as yet observed on spacecraft. We also assess the effect of optical pumping by earthshine and sunlight of both reaction products and effluents.

INTRODUCTION

The optical "foreground" contamination from gases excited by the interaction with the atmosphere of orbiting vehicles and re-entry bodies (and some sounding rockets), and from molecules inadvertently outgassed and purposefully exhausted from these spacecraft, falls into the three broad categories listed in Table 1. Figure 1 schematizes these UV-visible-IR radiations, and identifies the currently-perceived principal emitting species.

EXCITATIVE REACTIONS OF MOLECULES IN THRUSTER ENGINE EXHAUST

The velocities of the combustion products of typical control rocket engines are as high as 11-12 km/s relative to the (1000K, 1 km/s mean thermal speed) oxygen atoms and nitrogen molecules that compose most of the atmosphere at low-earth orbital altitudes. Table 2 lists the center-of-mass kinetic energies in collisions of 3½ km/s expected exhaust species from present thruster types with stationary O and the order-of-magnitude less abundant (and also less reactive) N_2. (The energies of outgas species lie between those shown.) This energy is in principle available for overcoming potential barriers of exothermic chemical reactions and populating upper electronic, vibrational, and rotational states of the products, as well as for similarly exciting the exhaust molecules themselves.

Figure 2 is a flow chart of the procedure that we applied (1) in identifying from the many possible reactions those of the highest-mole fraction exhaust molecules that result in electronically excited products with radiative lifetimes less than about 1/10 second. (When the lifetime of the upper state is longer, the excited molecules spread over such long distances that surface radiances become below detection thresholds.) Reactions that fail to conserve spin angular momentum are less favored; T → E collisions generally have lower probability than excitative atom-interchange collisions; and the cross-sections for exothermic reactions (such as $NO + O \rightarrow NO_2^*$) generally decrease with increasing kinetic energy of the participants. The principal excitative reactions of thruster exhaust meeting the tests of Figure 2 appear in Table 3, and estimates of their cross-sections from an evaluation of literature sources appear in Reference 1.

In practice only a sparse laboratory database exists on excitative processes at the kinetic energies in Table 2, and ab initio theory is in general not reliable. An example of the uncertainties for even so simple a collision process as H_2O (X,000) + O → H_2O (ν1,ν2,ν3) + O can be seen in Figure 3. Reference 2 gives some recently measured cross-sections for vibrational excitation in collisions with 8 km/s O atoms of CO, CO_2, and CH_4, and for dissociations of these species with vibrational excitation of a product molecule.

EROSIVE / EXTRACTIVE REACTIONS

While the rates of erosion of several types of exposed spacecraft materials by oxygen atoms with orbital relative velocities have been measured (3), the extraction mechanisms are not well understood. Yields of infrared radiation from the vibrationally-excited CO, CO_2, and OH molecules produced from organic (and thin layer of contaminant-covered metal) surfaces by O bombardment have recently been reported to be in the 10^{-2}-10^{-1} range (4), and further laboratory measurements are expected shortly.

SURFACE-CATALYZED RECOMBINATION OF IMPINGING ATMOSPHERIC SPECIES

Visible and near-ultraviolet glows with radiances comparable with that of the natural nightglow have been found in the last few years to extend from the ram-directed surfaces of satellites in low earth orbits (see in particular Reference 5). These emissions appear to be a general property of spacecraft moving through the thermosphere and, by inference, of exo-reentry and launch vehicles also. Table 4 is a listing of the perceivedly highest-radiance glows induced at spacecraft surfaces. These emissions arise from chemiluminous recombination reactions of ambient atoms and molecules incident on windward exposed surfaces, where the gas densities are high (and may be increased by backscattering from the atmosphere at the lower orbital altitudes).

These radiations, with the exception of the vibrational cascades from NO and NO_2 (final four entries in Table 4, predicted but not yet observed), are tabulated in order of increasing emission wavelength (second column). The spatial extents of these glows (third column) were derived with the assumption that the excited species are thermally accommodated with the surface before effusing from it. The radiances in the molecular bands (fourth column) are in general no more than rough estimates, as the absolute excitation probabilities are in very few cases known (if any).

Two extremes of projections for passive measurement of the radiances of off-spacecraft glows, parallel and perpendicular to the recombination surface are illustrated in Figure 4. If the distance L over which the molecule emits most of its radiation is less than the characteristic linear dimension W of the surface, signal/noise will be higher in the first view projection. When the desorbed particles do not collide with the atmospheric gas this distance L is to a good approximation (mean normal velocity of the desorbate) x (radiative lifetime of its excited state(s)). Thus (as examples) electric dipole-permitted electronic transitions with typical lifetimes ~10^{-7} sec would take place over much less than a millimeter from the excitation surface, and thus would reflect the recombination characteristics of the components of the optical sensor; transitions from NO_2 ($^2B_1$, $^2B_2$) with their ~100 msec lifetimes would produce radiation over a few cm, as observed; and (an extreme case) O_2 in its 3700-sec lifetime a $^1\Delta_g$ state would radiate over pathlengths of more than a thousand km when these molecules desorb into the zenith hemisphere, and thus exhibit extremely low surface brightness. When L < W a "van Rhijn" gain can be achieved on optically-thin glows by use of projections of narrow fields of view near-parallel to extended recombination surfaces. When L > W, even if the internally-excited species is desorbed non-isotropically the radiance of its glow in most projections varies with L^{-2} (neglecting collisional depopulation, or upward pumping of the desorbate stream), that is, with the inverse square of its radiative lifetime. For example if the 12-sec lifetime O_2 (b) and 3700-sec O_2 (a) states generally accepted as resulting from recombination of O atoms are populated at equal rates, column brightnesses in the [assumed unquenched] a → X Infrared Atmospheric bands would be a factor 10^5 lower than those in the b → X Atmospheric bands.

Both astronomical and terrestrial-atmospheric optical backgrounds are always present, and appear also as scattering off structures lying in sensor fields of view (over and above the infrared thermal radiation from such structures). Noise due to the natural variability of these backgrounds is a limiting factor in detection of all off-surface glows; the longer-lived recombination radiations listed in

Tables 5 and 6 would provide inadequate signal above this noise no matter what projection, or instrument throughput, is brought to bear. This effect is included in Figure 4's flow chart for excitation and detectability of recombination glows. A reaction with desorption in an excited stated must be favored by the substrate; indeed, visible spacecraft glow has been found to be less intense off exposed polyethylene $(CH_2)_n$ and anodized aluminum than off paints that have unsaturated chemical bonds (compare the low rate of recombinations and deactivations on teflon $(CF_2)_n$ in the laboratory).

With the assumption that its collision with the exposed surface does not electronically excite the impinging orbital-atmospheric atom or molecule, the upper electronic states of product molecules of surface-catalyzed recombination can be identified from their standard potential diagrams. (Too few initially-excited species are present in the flow to the surface to result in significant recombination radiation.) The thus-accessible ultraviolet- and infrared-radiating states from reactions of $O(^3P)$, $N(^4S)$, and $NO(X^2P)$ are shown in Tables 5 and 6, which include information from a survey of the literature on excitative recombination at laboratory surfaces (5). The most recently identified radiative state--also populated by gas-phase recombination--is N_2 $(a^1\Pi_g)$ (6).

As noted above, only a small fraction of these optical features would be detectable by passive spectroradiometry against the celestial and terrestrial backgrounds. Several of the other upper states lend themselves to detection by laser-probing methods, as we have suggested previously (5,7).

The database on observations of these spacecraft surface-catalyzed recombination glows is reviewed in a separate presentation at this meeting (Paper E1-3; refer also to the report of a recent conference devoted to this topic (8)). The by far dominant feature at the readily experimentally-accessible visible wavelengths is the NO_2 ($^2B_1$,$^2B_2$ $\rightarrow$ X^2A_1) pseudocontinuum from reaction of O atoms with NO molecules (naturally present in the atmosphere, or as has been postulated formed at the surface by recombination of atmospheric N and O). While the emission--the "Lewis-Rayleigh afterglow"--is a well-known feature of both laboratory gas discharges and the natural lower-thermospheric airglow (which forms under "radiation-stabilized" conditions, and therefore peaks well to the blue of the surface recombination glow) (see Reference 5; in particular Table 3), the off-surface emission has only been recently--in response to the stimulus of optical data from spacecraft--been unambiguously identified in the laboratory (9,10).

REFERENCES

1. Kofsky, I. L., Barrett, J. L., Brownrigg, T. E., McNicholl, P. N., Tran, N. H., and Trowbridge, C. A., "Excitation and Diagnostics of Optical Contamination in the Spacecraft Environment," AFGL-TR-88-0293, Air Force Geophysics Laboratory, Hanscom AFB, Massachusetts, July, 1988.

2. Green, B. D., and Caledonia, G. E., "Laboratory Investigations of High Velocity Atom Interactions," PAPER III-2, Johns Hopkins Univ./ Applied Physics Laboratory Vehicle-Environment Interaction Workshop, Laurel, Maryland, February 21-22, 1989.

3. Leger, L. J., and Visentine, J. T., "A Consideration of Atomic Oxygen Interactions with the Space Station," J. SPACECRAFT ROCKETS, New York, Vol. 23, 1988, pg. 505.

4. Fraser, M. E., Holtzclaw, K. W., and Gelb, A., "Mechanisms of High Velocity Oxygen Atom Reactions with Material Surfaces," PAPER SM21A-03, AGU Conference abstract publ. in EOS, Vol. 69, No. 44, November, 1988, pg. 1377.

5. Kofsky, I. L., and Barrett, J. L., "Spacecraft Glows from Surface-Catalyzed Reactions," PLANET. SPACE SCI., Vol. 34, No. 8, 1986, pp. 665-681.

6. Kofsky, I. L., "Excitation of N_2 Lyman-Birge-Hopfield Bands Emission by Low Earth Orbiting Spacecraft," GEOPHYS. RES. LETT., Vol. 15, No. 3, March, 1988, pp. 241-244.

7. Kofsky, I. L., and Barrett, J. L., "Spacecraft Surface Glows," NUCL. INSTR. METHODS IN PHYS RES., Vol. B14, 1986, pg. 480.

8. Meng, C. I., Ed., "Vehicle-Environment Interaction Workshop," Proc. of Johns Hopkins Univ./ Applied Physics Laboratory Vehicle-Environment Interaction Workshop, Laurel, Maryland, February 21-22, 1989.

9. J. A. Halstead, Trigge, N., Chu, A.-L., and Reeves, R. R., "Creation of Electronically Excited States by Heterogeneous Catalysis," in GAS-PHASE CHEMILUMINESCENCE AND CHEMIIONIZATIONS, Ed. A. Fontijn, Elsevier, Amsterdam, 1985, pp. 307-326.

10. Chu, A.-L. ; Reeves, R. R., and Halstead, J. A., "Surface-Catalyzed Formation of Electronically Excited Nitrogen Dioxide and Oxygen," J. CHEM. PHYS., Vol. 90, 1986, pp. 466-471.

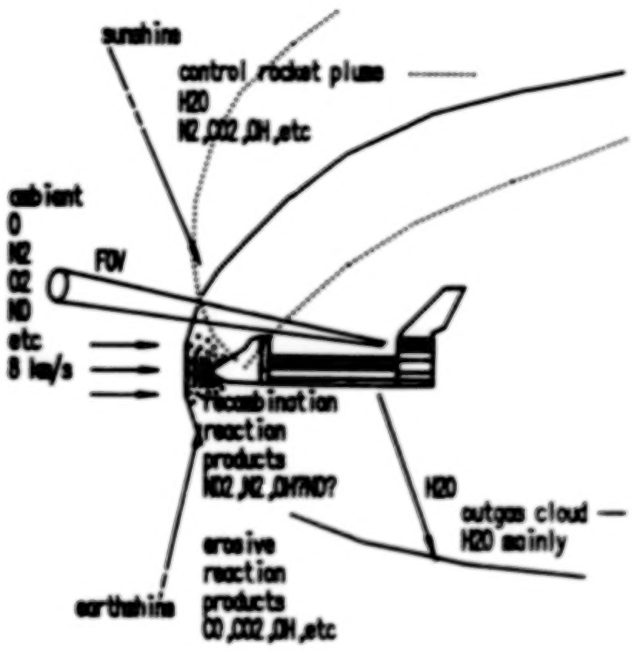


Figure 1. Sources of Contaminant Optical Radiations

Table I. Gaseous Optical Foregrounds from Spacecraft

LUMINESCENCE PROCESS	COMMENT
Gas-phase reactions between high kinetic energy desorbate/exhaust and atmosphere	Many reaction processes, with little quantitative guidance from experiment Large, separate topic
Surface-catalyzed recombination of atmospheric species	Multiple obeervations from space vehicles
Erosive/extractive reactions of incident species with exposed surface materials	Some laboratory data are now becoming available
Sunlight and earthshine pumping of gas cloud	Some evidence

Table II. Available Relative Kinetic Energy in Collisions with Exhaust Species.

Calculated for O :

Species	μ, amu	E_{max}, eV	E_{min}, eV
H_2O	8.47	5.74	0.45
N_2	10.2	6.91	0.55
H_2	1.78	1.20	0.092
CO	10.2	6.91	0.55
CO_2	11.7	7.92	0.62
S	0.941	0.64	0.050
NH_3-NO_3	14.0	9.48	0.74
O_2	10.7	7.25	0.57
OH	8.27	5.60	0.44
O	8.00	5.42	0.43
NO	10.4	7.04	0.55
N_2H_2	10.4	7.04	0.55
NO_2	11.9	8.06	0.63
NH_3	11.9	8.06	0.63
CH_2	7.47	5.07	0.39
CH_2S	11.5	7.79	0.61

Calculated for N_2 :

Species	μ, amu	E_{max}, eV	E_{min}, eV
H_2O	11.0	7.45	0.59
N_2	14.0	9.48	0.74
H_2	1.97	1.27	0.099
CO	[illegible]	9.48	0.74
CO_2	17.3	11.6	0.91
S	0.96[illegible]	0.66	0.052
NH_3-NO_3	22.3	15.2	1.19
O_2	14.9	10.1	0.79
OH	7.68	5.20	0.41
O	7.47	5.06	0.40
NO	14.5	9.82	0.77
N_2H_2	14.5	9.82	0.77
NO_2	17.4	11.8	0.93
NH_3	17.4	11.8	0.93
CH_2	9.33	6.33	0.35
CH_2S	16.6	11.3	0.89

E_{max} / E_{min} : 3½ km/s added/subtracted to 8 km/s orbital velocity

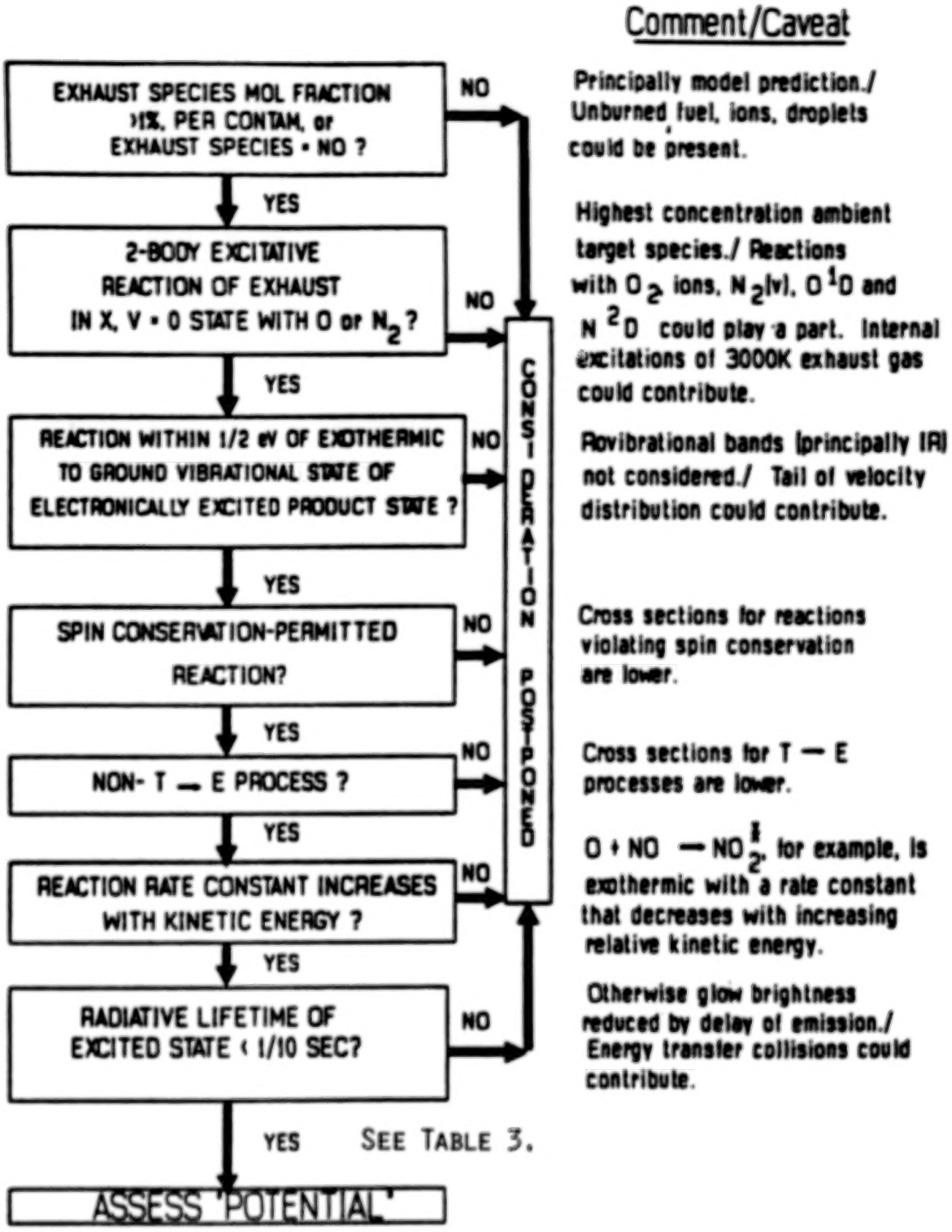


Figure 2. Potential Foregrounds from Electronically Excitative Reactions from Exhaust Species

Table III. Exhaust gas reactions meeting the tests illustrated in Figure 2.

Reactants	Products	E_{th}, eV	E_{max}, eV	$\sigma(E_{max})$, Å^2†	Emission System †††
$N_2O(X) + O(^3P)$	$OH(A)^* + OH(X)$	4.83	5.74	0.19	~2400-3500Å
$CO_2(X) + O(^3P)$	$O_2(A)^* + CO(X)$	6.67	7.92	0.10	Herzberg I, ~2400-4900Å
	$O_2(B)^* + CO(X)$	8.45			Schumann-Runge, <1900Å
$NO(X) + O(^3P)$	$O_2(A)^* + N(^4S)$	6.07	7.04	0.06	Herzberg I, ~2400-4900Å
	$O_2(A)^* + N(^4S)$	7.85			Schumann-Runge, <1900Å
$CO(X) + N_2(X)$	$CN(A)^* + NO(X)$	≥7.70††	9.48	-----	Red, ~4400-15000Å
$CO_2(X) + N_2(X)$	$NO_2(A)^* + CN(X)$	≥10.75††	11.6	-----	Yellow-green-IR continuum
	$CN(A)^* + NO_2(X)$	≥10.04††		-----	~4400-15000Å

† Estimated as described in the text.
†† These figures are the endothermicities E_0 of the reactions. No information on the barrier threshold is available.
††† To ground state (X) of the excited molecule.

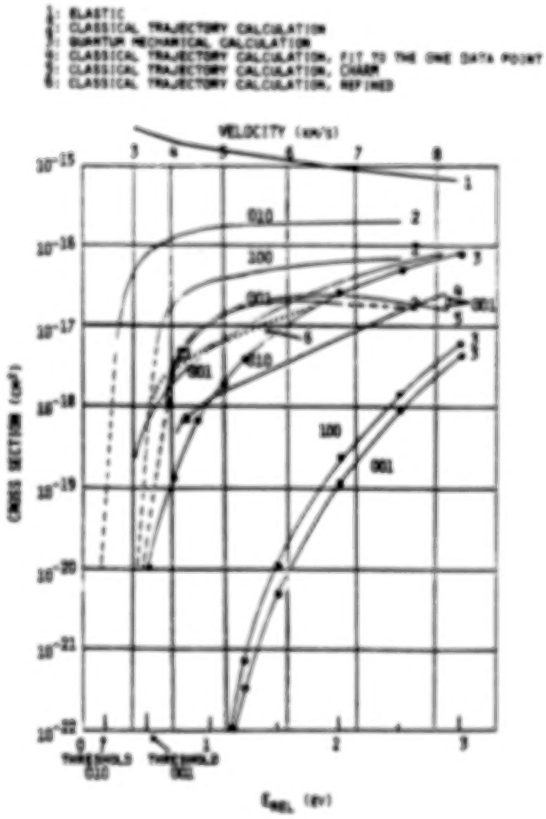
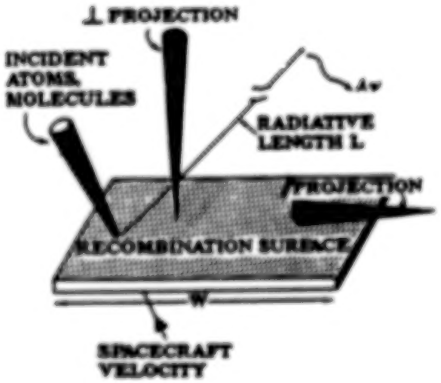


Figure 3. V-V in $O^3P + N_2O(X)$ collisions

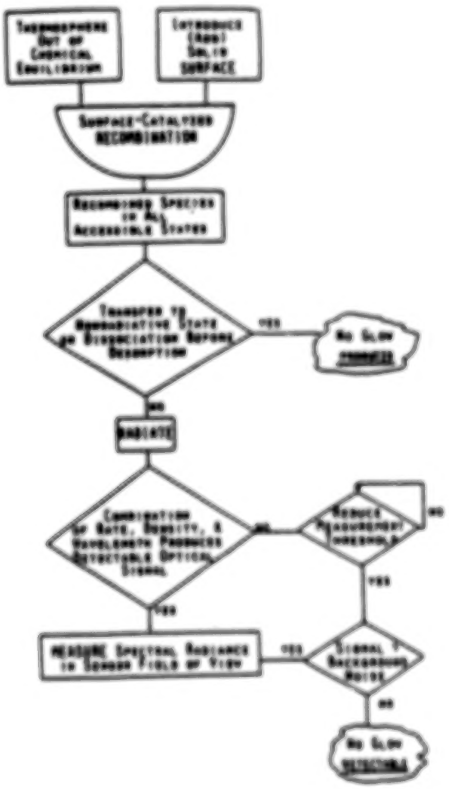
Table IV. Principal Spacecraft Surface-Catalysed Recombination Glows

Band System	Δλ	Spatial Extent*	Radiance**	Comments	Ref
N_2 Lyman-Birge-Hopfield ($a \rightarrow X$)†	1200-2100Å	4 cm	2 kR	Strong dependence on altitude unexplained	6
NO Beta ($B \rightarrow X$)††	2000-3800Å	1½ mm	1 kR	Excited at exposed sensor surfaces; $A \rightarrow X$, $C \rightarrow X$ not observed	3
O_2 Herzberg I ($A \rightarrow X$)	2500-4900Å	60 m	2 kR	Other recombinant O_2 states produce weak emission radiances	1
NO_2 pseudo-continuum ($^1B_1, {}^1B_2 \rightarrow {}^2A_1$)†	3900Å-2½ μm	5-10 cm	10 kR	"Shuttle glow" from NO + O	5
N_2 Vu-Benesch ($B \rightarrow B$), First Positive $B \rightarrow A$††	1-3 μm & 5000-10000Å	4-40 cm (4½ cm)	1 kR each	First Positive cascade follows $B \rightarrow B$; $A \rightarrow X$ weak; other recombinant N_2 features weak also	1
NO $\Delta v = 2$	2.7-3 μm	100 m	4×10^{-13} w/cm^{2}ster	Vibrational cascade follows $B \rightarrow X$, $v>0$	3
NO $\Delta v = 1$	5.3-6 μm	100 m	2×10^{-13} w/cm^{2}ster	Same as $\Delta v = 2$	3
NO_2 $\nu_1 + \nu_3$	3.4-3.8 μm	2½ m	3×10^{-11} w/cm^{2}ster	Vibrational cascade follows X, ν_1, ν_2, ν_3	3
NO_2 ν_3	6.0-6.6 μm	2 m	4×10^{-10} w/cm^{2}ster	Strongest cascade component of NO_2	3

* (mean effusive desorption velocity)*(radiative lifetime of upper excited state).
** Nominal planning estimate, perpendicular projection, complete band system at 225 km altitude.
† Identified from spacecraft. †† Possible identification.



a) View Geometry



b) Conditions for production and detection of spacecraft off-surface glows

Figure 4. Conditions for Spacecraft-Induced Glows

Table V. UV Recombination radiations at low earth orbital altitudes

Recombinant Species	Upper Electronic States Accessible,* lifetime	Ultraviolet Band Systems Radiated	Radiative Pathlength**
$O^3P + O^3P \rightarrow O_2^*$	$\underline{A}^3\Pi_u^+$, 0.18 sec A'(or C)$^3\Delta_u$, ~20 sec c, a, b	Herzberg I ($\rightarrow$X) Herzberg II (Visible and IR)	80 m 10 km 5 - 100's km
$N^4S + N^4S \rightarrow N_2^*$	$a^1\Pi_g$, 120 μsec $A^3\Sigma_g$ with v = 20, ~1000 μsec $A^3\Sigma_g$ with v = 5, 2 sec $B^3\Pi_g$ a',B',W ~0.04, 10^{-5} and 10^{-3}-10^{-2} sec	L-B-H ($\rightarrow$X) ($\rightarrow$B, IR; B$\rightarrow$A(v=5), Vis & IR) Vegard-Kaplan Visible, but produces A, low v Mostly IR	6 cm 1 km
$N^4S + O^3P \rightarrow NO^*$	$\underline{B}^2\Pi$, 3.2 μsec A, 0.2 μsec C, 0.025 μsec a, b, 0.16 and 6 sec	β ($\rightarrow$X) γ ($\rightarrow$X) δ ($\rightarrow$X) (Visible and IR)	1¼ mm 1/10 mm
$NO + O \rightarrow NO_2^*$	$^2B_1$, $^2B_2$, 25-250 μsec	Pseudocontinuum, no UV component	1-10 cm

* Those underlined have been detected in the laboratory.
** Assuming the desorbate molecules have equilbrated-effusive desorption velocites.

Table VI. Infrared (≥ 8000 Å) Surface Recombination Species

Excited Molecule	Source	Comments
NO ($X^2\Pi$, v > 0)	Cascade from NO ($B^2\Pi$)	Present if β bands are excited, 2.7 and 5.3 μm
$NO_2(X^2A_1, \nu_1\nu_2\nu_3)$	Cascade from $^2B_{1,2} \rightarrow X,v$	Several band systems, strong; 3.6 and 6.2 μm, extending to 13 μm
NO_2 ($^2B_1$, $^2B_2$)	NO + O Recombination	Long-wavelength tail of pseudo-continuum extends to ~3 μm
$N_2(W^3\Delta_u)$	N + N Recombination	Inferred from laboratory results; Wu-Benesch (W$\rightarrow$B) bands
$N_2(B^3\Pi_g)$	Cascade from W (or A) state	Seen in the laboratory; First Positive bands (B$\rightarrow$A) extend into the near-IR
$O_2(a^1\Delta_g)$	O + O	IR Atmospheric bands (a$\rightarrow$X) 1 hour radiative lifetime results in extremely low brightness
$O_2(b^1\Sigma_g^+)$	O + O	Near-IR Atmospheric bands (b$\rightarrow$X) 12 sec radiative lifetime results in low brightness
$NO(b^4\Sigma^-)$	N + O	Near-IR Ogawa (b$\rightarrow$a) bands; 6 sec radiative lifetime
$O_3(\nu_3)$	O + O_2	Not identified in laboratory; improbable, but if excited 001-000 at 9.6 μm, + hot bands

CONTAM 3.4 (SDI APPLICATIONS VERSION): STATUS REPORT

R. Hoffman
Science Applications International Corporation

(Paper not provided by publication date.)

THE ALTITUDE AND ANGULAR DEPENDENCE OF RADIATION CONTAMINATION

J. Elgin and M. Pervaiz
Spectral Sciences, Inc.

D. Cooke and E. Murad
Geophysics Laboratory, Hanscom AFB

M. Tautz
RADEX, Inc.

(Paper not provided by publication date.)

SPACECRAFT EXTERNAL MOLECULAR CONTAMINATION ANALYSIS

H. K. F. Ehlers
National Aeronautics and Space Administration
Johnson Space Center, m/c ES531
Houston, Texas 77058

ABSTRACT

Control of contamination on and around spacecraft is required to avoid adverse effects on the performance of instruments and spacecraft systems. Recent work in this area is reviewed and discussed. Specific issues and limitations to be considered as part of the effort to predict contamination effects using modeling techniques are addressed. Significant results of Space Shuttle missions in the field of molecule/surface interactions as well as their implications for space station design and operation are reviewed.

INTRODUCTION

Contamination and the resulting performance degradation of systems generally is of considerable concern to organizations which design, build and operate spacecraft. Contamination can have a variety of causes and effects. This presentation addresses spacecraft on-orbit contamination due to the movement of neutral molecules only. Neutral molecules may, when present in the path of light, affect the quality of optical observations of objects in space—objects such as the sun, planets, stars, other spacecraft, etc. They may also, when deposited on surfaces, reduce the quality of mechanical, electrical and radiative properties of hardware surfaces, e.g. mirrors, lenses, windows and thermal control surfaces. Sources of this contamination may be spacecraft hardware exposed to the vacuum of space, zero gravity and impinging atmosphere (e.g. atomic oxygen), or they may consist of thrusters, gas vents, gas leaks, etc. Scientific instruments as well as spacecraft subsystems share in the production of contaminants on the one hand and in the adverse effects caused by these contaminants on the other. In order to minimize problems due to contamination, a very specific contamination control plan must be developed early during the conceptual design phase of the spacecraft. The contamination control plan affects the spacecraft in many ways. For instance, it affects the spacecraft configuration, design, choice of materials, and spacecraft operation. Therefore contamination control must be an integral part of the spacecraft development and operation and must be appropriately documented.

Specific areas of spacecraft design and operation which are affected by the contamination control plan are:

a) material selection/processing/control with regard to molecular outgassing;

b) design and performance of pressurized compartments, e.g. fluid containers and lines and line connections, with regard to gas leakage;

c) propulsion system design and performance with regard to molecular deposition; and

d) gas venting methods/procedures and fluid management system design and performance with regard to flow rate limitations.

Also affected are all aspects of operations on the ground and during launch as well as planning of protective measures for instruments.

The efforts to control contamination on spacecraft include, in addition to creating a contamination control plan, the definition of contamination requirements, the development of predictive models, and measurement of contamination levels to verify the requirements. The contamination control plan itself describes organization, methods, procedures and controls to be applied in order to meet the contamination requirements.

For the kind of contamination discussed here the contamination requirements define the maximum levels of external induced neutral molecular environment permitted, to ensure that maximum utilization of spacecraft capabilities is not restricted by contamination effects. This has generally led to the establishment of contamination level limits for the following categories:

a) background spectral irradiance (including "spacecraft glow"),

b) molecular column density, and

c) molecular deposition.

Such requirements, among others, are contained in Space Shuttle as well as Space Station Freedom documents, for instance. Although the background spectral irradiance is the dominant concern for light observations, molecular column density limits also are specified based on their close relationship to the background radiation as well as their direct dependence on spacecraft hardware and operational aspects.

Explanatory information on the terms "column density" and "deposition" cited in requirements is provided in Figures 1 and 2, using as an example a space station with payload at the prime measurement point (PMP) location.

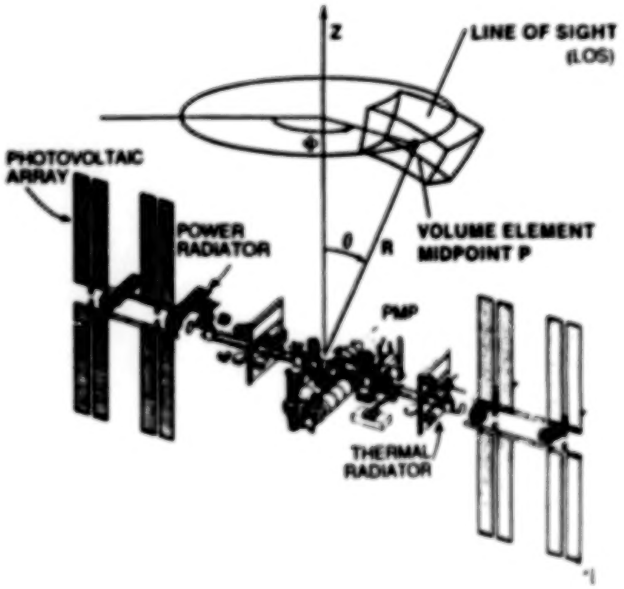


Figure 1. Elemental volume geometry used for density and column density calculations.

Figure 1 shows a finite volume element representing a locally constant (but dynamic) gas density. Imagining a sequence of adjacent elements like this along a straight line beginning at the PMP and ending at infinity, called a line-of-sight (LOS), one can add all products D•L of element density (D) and element segment length (L) (or integrate density over distance) and arrive at the total value defined as column density.

Figure 2 demonstrates two different molecular flow mechanisms leading to potential contamination deposition. Figure 2a explains typical direct source-to-receiver flow and Figure 2b depicts potential depo-sition resulting from "return flux" of contamination molecules due to collisions of "departing contamination molecules" with either ambient or other departing molecules within a specified field of view (FOV) originating at the PMP.

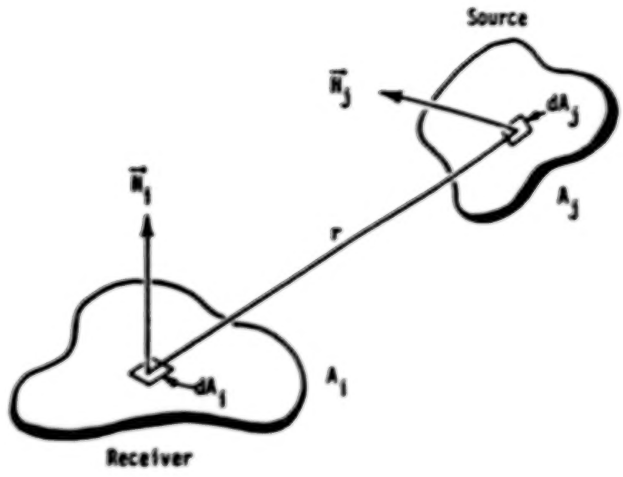


Figure 2a. Geometry for configuration factor between finite areas used for direct flux calculations.

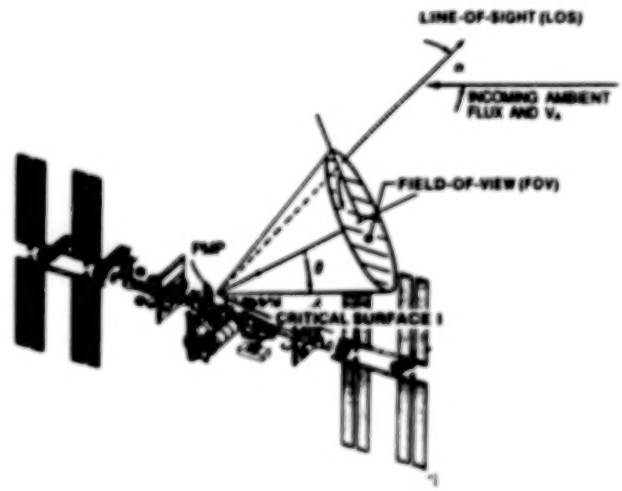


Figure 2b. Critical surface location and field-of-view used for return flux calculation.

Predicting the external induced environment —based on spacecraft design and planned operation before any measurements on the final product can be taken— is an important part of the task of controlling contamination. This is the purpose of models. The results of modeling are used to determine whether a specific design will be adequate to meet the requirements or whether changes to the design and/or operational plans are necessary. Therefore a model which fulfills the needs of the program must be available. Valid models can be very simple and relatively inexpensive, or they may be extremely complex and costly. The information provided in this presentation is intended to aid in the process of planning and selecting a model which is adequate for the task at hand at reasonable cost. Recent developments in modeling techniques— with modeling results as well as measurements affecting model design— are discussed using examples from the Space Shuttle and Space Station Freedom programs.

EXTERNAL INDUCED ENVIRONMENT MODELING

Assuming an ideal scientific approach, a math model should represent an accurate description of the physical and chemical processes which lead to the observed environment. A review of the induced environment data base indicates that many processes involved are not understood well enough to be accurately formulated. One reason for this is the lack of measured data relating to reactions of ambient molecules approaching the spacecraft at orbital velocities. This means that the goal of a perfect model for the induced environment cannot be met at the present time, especially with regard to large spacecraft such as the Space Shuttle and the Space Station Freedom. Gathering of the necessary data by the scientific community will take a long time and large sums of money, since most measurements must be made from spacecraft in earth orbit. In the meantime valid models are needed to support current development of spacecraft such as the Space Station Freedom. These developments generally must take place "on schedule" with very limited time and with funds which do not allow for scientific research to improve the state of the art of modeling. Therefore present development of analytical models to support spacecraft design is restricted by the available data base. Sometimes this leads to a level of model output uncertainties which is considered to be unacceptable by the scientific community. Nevertheless these models, which predict the results of events which generally have not been measured before, seem to serve the purpose for which they were developed. However the user should be aware of the limits in accuracy. Models which go into a lot of detail tend to become so complex that only a few experts can understand and use them correctly. Additional issues raised concerning models are transportability, software language, user friendliness and interfacing with other models. General solutions are costly. Simplified models (with limited accuracy) and/or charts, tables, plots, etc. created with complex models may be more helpful to the user to complete his task.

Models which predict molecular spacecraft contamination must produce output that corresponds to parameters of the specific requirements. These are density, column density, and deposition due to direct and return flux. Models must include —but need not go beyond— the total environmental conditions (ambient and induced) actually to be encountered.

Program input includes gas kinetics (formulation of processes), geometric configurations, spacecraft trajectory/attitude, ambient environment, instrument FOV/direction, etc. Present model prediction uncertainties are dictated primarily by uncertainties about applicable values for a number of input parameters, specifically:

- molecular collision frequencies (and molecular collision cross sections);
- inelastic molecular collision cross sections and reaction probabilities (excited molecules, chemical surface reactions, release from the surface of molecular species other than those arriving);
- surface accommodation coefficients;
- molecular surface deposition/re-evaporation rates;
- material outgassing rates (long/short term);
- pressurized system leakage locations and distribution of rates; and
- time dependent atmospheric density composition and molecule velocity distributions.

The errors in the input arguments propagate to corresponding errors of model output data. Some of them are significantly larger than errors due to modeling technique shortcomings. The user must recognize this when interpreting predicted data, comparing outputs of different models, or reviewing modeling techniques. The existence of significant uncertainties in input data and resulting predictions also points toward the importance of performing measurements needed to reduce error margins.

Two principal methods are currently in use to model the induced molecular spacecraft environment: a discrete particle method and a gas continuum method.

The Direct Simulation Monte Carlo (DSMC) approach developed by G. A. Bird[1] is generally the basis for the discrete particle technique. This method is known for having produced data in good agreement with measured data concerning a number of gas flow problems, specifically molecular distributions created by spacecraft thrusters. The disadvantage of using this method lies in very long computer run times as well as the requirement of extensive experience in using the codes and dealing with the statistical errors. Only a very few organizations have the expertise necessary to utilize this approach.

In an effort to find simplified as well as user-friendly formulations for use in general modeling, gas continuum methods have been developed. They are based on solutions of the Boltzmann equation. The disadvantage is that this equation can generally only be solved by applying numerical techniques. These generally observe the laws of physics but introduce approximations (and with them certain errors) necessary to find acceptable solutions. The range of application is limited to the validity range of the equations used and the assumptions made to simplify the solution. The greatest advantages are ease of use and reasonably short computer run times. The limits in their range of application can be overcome by skillful combination with well-formulated results produced by other methods such as DSMC, Method of Characteristics (MOC), or even measurements.

An example of this is the model developed for Space Station Freedom (and Space Shuttle) application: MOLFLUX[2]. It can deal with numerous nodes. Direct molecular fluxes (including surface reflections) calculated by the model are based on geometric configuration factor data combined with source emission characteristics. Direct deposition fluxes, densities and column densities are derived from these molecular fluxes, including ambient atmospheric fluxes.

Backscattering return flux/deposition predictions are based on a numerical integration of the Bhatnagar-Gross-Krook (BGK) model approximation of the Boltzmann equation for gas mixtures[3] developed by Robertson.[4] Equations describing the fluxes from concentrated (high density) sources such as thrusters and gas vents are formulated either from measurement results or from DSMC and MOC output. This method leads to adequate predictions at reasonable expense in time and cost for spacecraft as large as a space station.

RESULTS OF MEASUREMENTS AND PREDICTIONS (EXAMPLES)

A few examples of significant results of flight measurements and model predictions will be discussed now to demonstrate the impact of the environment on spacecraft, specifically the Space Shuttle and the Space Station Freedom.

Ambient Ram Effect on Space Shuttle

The first example illustrates and quantifies the increase in density above ambient atmospheric levels on the ram side of a surface moving in space at orbital velocity. The increase is expected according to the kinetic gas theory and has been a factor in atmospheric measurements since the early days of space flight. But it also affects the operation of instruments in the Space Shuttle payload bay and on the Space Station Freedom

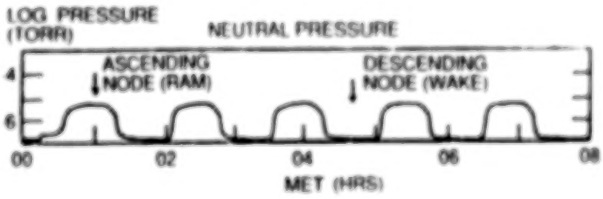


Figure 3 Pressure measured during the STS-3 mission as a function of time. Space Shuttle altitude = 240 km. Ambient density = 5×10^9/cm^3.

Figure 3 shows a result of measurements performed by Shawhan *et al.*, using an ion gauge on the Plasma Diagnostic Package (PDP) during the STS-3 mission[5]. The pressure measured on the third mission day, mission elapsed time (MET) 0 to 8 hours, varies periodically by approximately two orders of magnitude with the orientation of the payload bay relative to the flight direction (ram—wake) and therefore cannot be the result of outgassing in the bay. Maximum pressure was less than 1×10^{-5} torr, equivalent to a gas density of approximately 3×10^{11} molecules/cm^3 at ram orientation of the Space Shuttle payload bay. The Space Shuttle flew at 240 km altitude. The total neutral (undisturbed) ambient density was determined to be 5×10^9 molecules/cm^3 using the MSIS-83 model under conditions existing at that time and place. A comparison of these densities indicates that the density in the bay increased almost by a factor of 60 with respect to ambient.

The MOLFLUX model has been used to determine this density increase for comparison. In concept, the calculations included reflection of ambient molecules by the payload bay walls into random directions up to four times. Also, complete accomodation and conservation of fluxes for individual species on surfaces was assumed. The MOLFLUX model-predicted total density distribution along an LOS from the bottom of the empty payload bay outward is shown in Figure 4a.

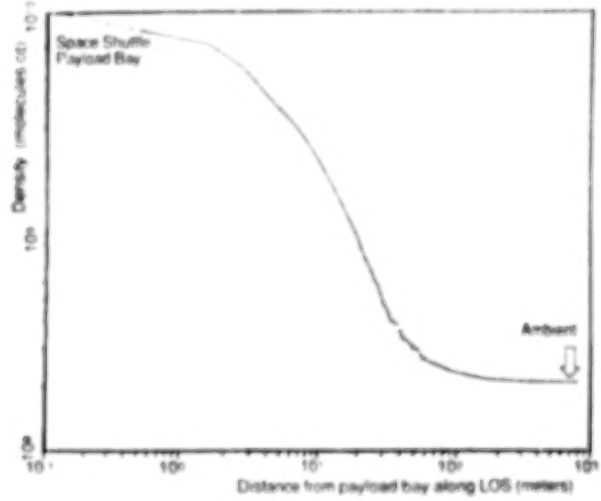


Figure 4a. Density distribution along an LOS from the bottom of the space shuttle payload bay outward in the ram (+Z) direction predicted by the MOLFLUX model.

For large distances from the bay the ambient density is assumed to be 2×10^8 molecules/cm^3 (74.6% O, 23.8% N_2, 1.6% O_2). The calculated density near the bottom of the bay reaches values above 1×10^{10} molecules/cm^3, approaching the ratio above ambient measured in a bay partially occupied by payloads. Apparently the payload bay itself almost acts like the enclosure of a pressure gauge. For a more realistic analysis, however, physical/chemical molecular reactions at the surfaces must be considered, provided they are known. Figure 4b shows the geometry involved in these calculations.

Ambient Ram Effect on Space Station Freedom

Density increases are expected also to occur on the ram

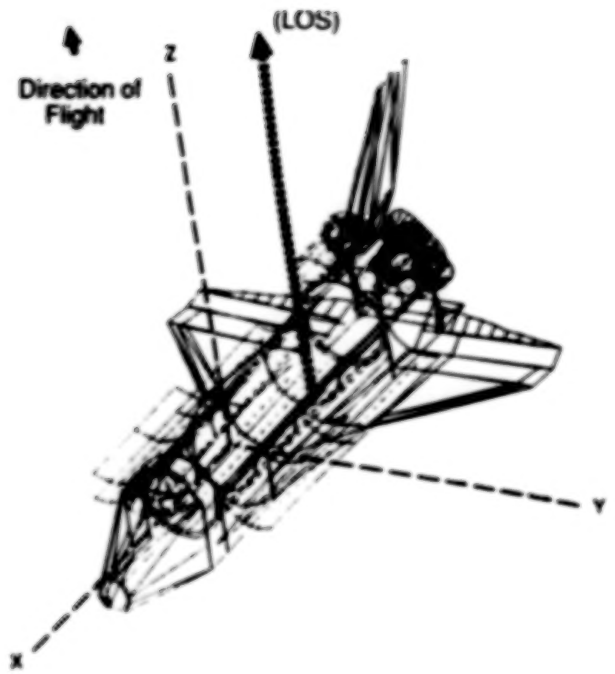


Figure 4b. Geometry used for density calculations (Space Shuttle).

side of Space Station Freedom surfaces, particularly the large solar panels. The effects of these higher densities must be studied very carefully. The higher densities may lead to enhanced plasma.

The overall effect on the density near the Space Station Freedom and the column densities, calculated using the MOLFLUX model and the ambient molecular reflection concept described above, is illustrated in Figure 5 and Table I.

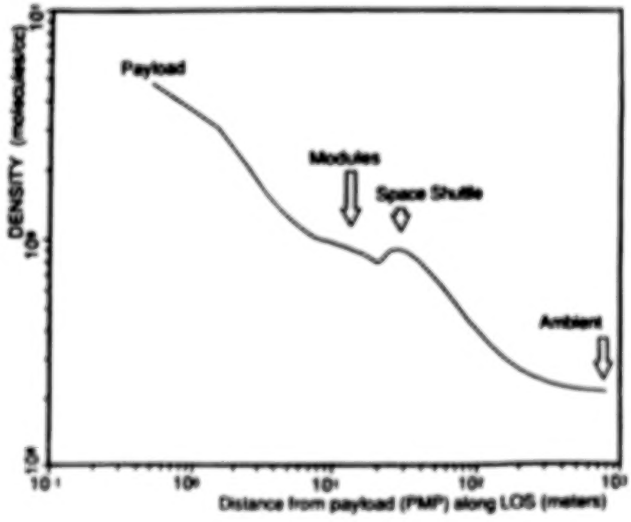


Figure 5a. Density distribution along an LOS from the PMP on the Space Station Freedom in the ram direction predicted by the MOLFLUX model. Module leakage = 5 lb/day. Uniform outgassing rate = $1x10^{-11}$g/cm^{2}sec. Ambient density = $2x10^8$/cm^3.

In Figure 5a the total density is plotted against distance along an LOS originating at a centrally located payload on the main truss with a +X (ram) direction. See Figure 5b. In addition to the ambient gas flow (at $2x10^8$ molecules/cm^3), gas sources such as module leakage (5 lb/day) and uniform material outgassing ($1x10^{-11}$ g/cm^{2}sec) are contributing to the result.

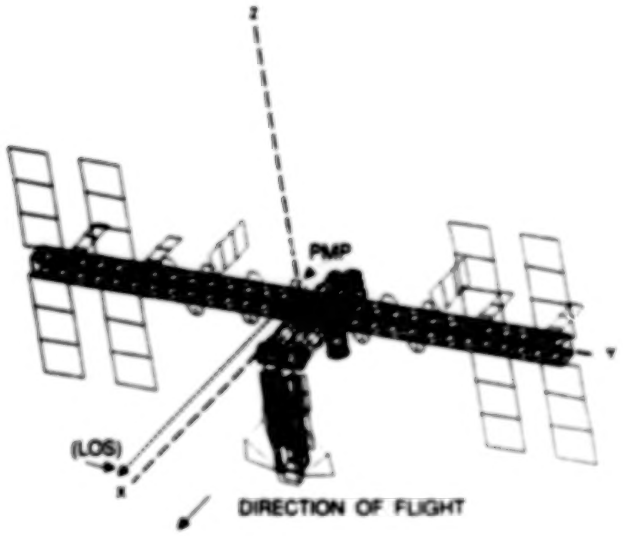


Figure 5b. Geometry used for density calculations (Space Station Freedom).

Local density increases due to ambient flux impinging on the payload (represented by a disk located at the origin of the LOS), Space Station Freedom modules and the Space Shuttle are indicated. They generally become more significant near the ramming surfaces.

Table I. Column densities in the ram (+X) direction. Module leakage = 5 lb/day. Uniform outgassing rate = $1x10^{-11}$g/cm^{2}sec. Ambient density = $2x10^8$/cm^3.

CONFIGURATION	Out Gas	H_2O	CO_2	TOTAL (ALL)
Space Station Freedom (SSF) + Payload (P/L) + Truss + Space Shuttle (SSh)	$9.6x10^{9}$	$1.3x10^{10}$	$9.3x10^{9}$	$7.6x10^{12}$
SSF + P/L + SSh	$8.1x10^{9}$	$1.3x10^{10}$	$9.3x10^{9}$	$7.5x10^{12}$
SSF + P/L + Truss	$7.5x10^{9}$	$1.4x10^{10}$	$1.0x10^{10}$	$6.3x10^{12}$
SSF + P/L	$6.8x10^{9}$	$1.3x10^{10}$	$9.3x10^{9}$	$6.0x10^{12}$
Truss + P/L	$1.9x10^{9}$			$1.8x10^{12}$
Truss only	$6.0x10^{8}$			$3.0x10^{11}$

Table I summarizes the results of column density calculations under equivalent conditions for several different Space Station Freedom/Space Shuttle configurations. The "total" values listed include data for other gases, mainly O and N_2, which are not itemized separately in the table. Several conclusions are obvious.

The truss contributes a relatively insignificant amount to the overall total values (at the given outgassing rate). This means that the truss, due to its relatively small surface area as well as its low outgassing rate, can be practically ignored in comparison with the large area space station elements in predicting the induced environment. This result is very important as part of the modeling process, since ignoring the truss saves several hundred nodes and thus large amounts of computer time. In addition it is interesting to note that, for this LOS and uniform outgassing rate, the column density of outgassing molecules for the Space Station Freedom (with or without a docked Space Shuttle) has nearly the same value, about $1x10^{10}$ molecules/cm^2. Since the outgassing rate of Space Station Freedom material will significantly decrease with time in orbit, its contribution to the total outgassing column density will also decrease and the presence of the Space Shuttle will be relatively more noticeable than Table I indicates. The total column density of all species is dominated by the contribution from the induced ambient flux (about $1x10^{13}$ molecules/cm^2) with and without the docked Space Shuttle. Column densities for H_2O and CO_2 listed in the table are due to space station module leakage. Review of column density data for other LOS's in the X-Z plane (not shown) reveals that the values for total column densities decrease when the LOS's approach the +Z direction. In reality, values will differ somewhat from the results shown, depending on 1) real gas flow rates from all sources which are very time-dependent, and 2) surface reactions of ambient molecules and atoms, specifically atomic oxygen. Nevertheless the presented data provide valuable information about the criticality of the natural and induced neutral Space Station Freedom environment with regard to station design and planned operation.

Direct Flux to Space Station Freedom Elements

By far the largest contribution to the deposition of contaminants on surfaces is the result of direct flux from sources. A model such as MOLFLUX can efficiently calculate the values of direct contamination fluxes between all surfaces and from concentrated sources to surfaces. The output data must be carefully analyzed as to the fraction actually condensing on any surface depending on rates, temperatures and surface characteristics. For a spacecraft as large as the Space Station Freedom, with numerous surfaces and gas sources, the direct flux/deposition data bank becomes complex and huge in size. It is presently available (only on microfiche) and spares the users the effort to do their own modeling.

To summarize these data, it can be pointed out that very significant fractions (some larger than 10%) of outgassing molecules may impinge on certain payload and other surfaces despite relatively large distances from sources. This fact is due to the large area of some of the outgassing materials, namely the surfaces of the modules. The Space Station Freedom external contamination control requirements (defined in SSP-30426) limit the permitted flux of molecules emanating from the core space station such that "the mass deposition rate on two 300°K surfaces both located at the PMP with one perpendicular to the +Z axis and the other whose surface normal lies in the horizontal plane and at critical power locations with an acceptance angle of 2π steradian shall be no more than $1x10^{-14}$ g/cm^{2}sec (daily average)". Therefore materials spread over large external areas of the station should be outgassing at rates lower than $1x10^{-13}$ g/cm^{2}sec to meet these requirements. Characteristics of these materials must be carefully measured and controlled to precisely determine and limit their impact.

Molecular Deposition Analysis

Molecular deposition is generally very difficult to predict. The reason is the dependence on many parameters, especially time and varying material surface characteristics. Deposition is the result of balancing impinging and departing fluxes, varying with time. A typical result is shown in Figure 6, where amounts of deposition on several temperature-controlled quartz crystal monitors (TQCM) are plotted as functions of time during the period of Space Shuttle thruster operations named "L2U test" on mission STS-3.[6]

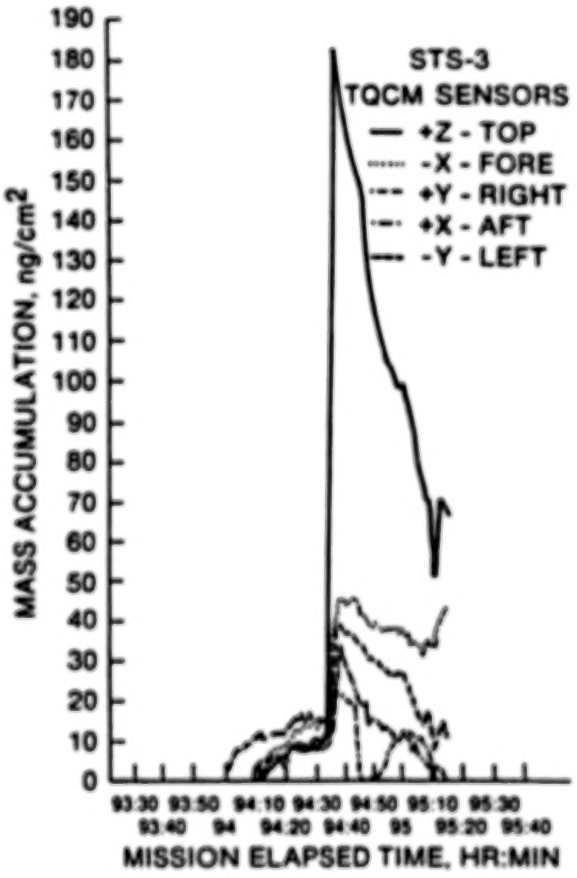


Figure 6. Mass accumulation on TQCM sensors during STS-3 L2U engine firing.

Part of this test consisted of the operation of three upward-firing reaction control system (RCS) engines for a period of about 100 seconds with only a few interruptions. A detailed description and analysis of the results regarding molecular deposition was presented previously.[7] Therefore only some of the analysis results are summarized here. Flux from the engines could not reach the TQCM's directly. Collisions of effluent molecules with ambient molecules as well as other effluent molecules must have been the primary process resulting in the deposits. The main contributor to the deposition was assumed to be $MMHNO_3$ with a vapor pressure sufficient to cause re-evaporation. The data analysis indicated that TQCM exposure to the ambient atomic oxygen flux accelerated the cleaning process. However, more precise measurements are needed to understand the deposition/re-evaporation mechanism.

Investigations performed in the meantime —and supported by MOLFLUX calculations— provided rates of Space Shuttle engine effluent fluxes on Space Station Freedom elements during Space Shuttle proximity maneuvers. Preliminary results show that levels of molecular deposition can meet requirements, provided that proximity operations are optimized for minimum flow of Space Shuttle thruster effluent toward the station. These maneuvers lead to molecular surface deposition which must be assessed as to the effect on Space Station Freedom and payload operations.

The extent of initial deposition, contamination cleanup and permanent deposition caused by bipropellant thrusters used on the Space Shuttle is of particular interest and has been a subject of continued investigations. Trinks studied the effects of monomethylhydrazine-nitrogen tetroxide bipropellant (MMH/NTO) thruster contamination on spacecraft materials in a small vacuum chamber[8]. He found deposition of droplets composed of a combination of H_2O, MMH and MMH-nitrates leading to nitric acid and MMH-polymerization products. Exposure to various environments resulted in some unidentified permanent residues.

The difficulties in performing such deposition/cleanup studies with reasonable results in a laboratory on Earth were recently experienced at Johnson Space Center (JSC). Material samples were exposed to effluent from a bipropellant thruster in a vacuum chamber at the White Sands Test Facility. After return to JSC the samples were placed into an asher, and alternatively into a flowing afterglow apparatus (oxygen discharge), to study ways to remove the brown hygroscopic deposits from the contaminated surfaces. The result was partial cleaning of the samples and a solid deposit identified as iron oxide. Considering the circumstances in the test chamber, particularly the difficulties in maintaining a clean chamber environment, the precise origin of the iron oxide in the contaminated sample could not be located.

It appears that conclusive deposition measurements involving thrusters must be performed using the actual system to be evaluated within the neutral space environment in order to produce meaningful results.

—o,o—

Other types of contamination effects have been observed and measured and are being analyzed at present with the expectation of incorporation into models. They are, however, beyond the scope of this brief review. Atomic oxygen erosion effects as well as "vehicle glow", for instance, fall into this category.

CONCLUDING REMARKS

Many data needed to develop models of contamination flow involving spacecraft have been measured recently. They have helped to significantly improve modeling accuracy and verification. Much work remains to be done, specifically flight measurements, to arrive at an even better understanding of the major processes influencing contamination deposition and effects on optical observations.

REFERENCES

1. Bird, G. A., MOLECULAR GAS DYNAMICS, Oxford University Press, London, 1976, pp. 118-132.

2. Rios, E. R., Moffitt, R. G. and Vinson, K. E., "MOLFLUX: Molecular Flux User's Manual," JSC-22496, U. S. National Aeronautics and Space Administration, Johnson Space Center, Structures and Mechanics Division, Materials Branch, Houston, Texas, March, 1987.

3. Bhatnagar, P. L., Gross, E. P., and Krook, M., "A Model for Collision Processes in Gases: I. Small Amplitude Processes in Charged and Neutral One-Component Systems," PHYS. REV., Vol. 94, No. 3, May, 1954, pp. 511-525.

4. Robertson, S. J., "Spacecraft Self-Contamination Due to Backscattering of Outgas Products," INTERIM REPORT LMSC-HREC TR D496676, Lockheed Missiles and Space Company, Inc., Huntsville, Alabama, January, 1976.

5. Shawhan, S. D., Murphy, G. B., and Pickett, J. S., "Plasma Diagnostics Package Initial Assessment of the Shuttle Orbiter Plasma Environment," JOURNAL OF SPACECRAFT AND ROCKETS, Vol. 21, July-August , 1984, pp387-391.

6. Miller, E. R., Ed., "STS-2, -3, -4 Induced Environment Contamination Monitor (IECM) Summary Report," NASA TM-82524, U. S. National Aeronautics and Space Administration, George C. Marshall Spaceflight Center, Huntsville, Alabama, February, 1983.

7. Ehlers, H. K. F., "An Analysis of Return Flux from the Space Shuttle Orbiter RCS Engines," AIAA-84-0551, AIAA 22nd Aerospace Sciences Meeting, Reno, Nevada, January 9-12, 1984.

8. Trinks, H., "The Effects of Bipropellant Thruster Contaminants on Spacecraft Materials," Fourth European Symposium on Spacecraft Materials in Space Environment, Centre Nationale d'Etudes et Recherches de Toulouse (CERT), Toulouse, France, September 6-9, 1988, pp. 285-296.

AN ASSESSMENT OF PROPULSION SYSTEM CONTAMINATION ON SPACE PLATFORM OPERATIONS

C. Maag
Science Applications International Corporation

(Paper not provided by publication date.)

AUTONOMOUS POWER EXPERT FAULT DIAGNOSTIC SYSTEM FOR SPACE STATION FREEDOM ELECTRICAL POWER SYSTEM TESTBED

Long V. Truong, Jerry L. Walters, and Mary Ellen Roth
NASA Lewis Research Center
21000 Brookpark Road, MS 301-3
Cleveland, Ohio 44135

Todd Quinn and Walter M. Krawczonek
Sverdrup Technology, Inc.
NASA Lewis Research Center Group
21000 Brookpark Road, MS 301-3
Cleveland, Ohio 44135

ABSTRACT

The goal of the Autonomous Power System (APS) program is to develop and apply intelligent problem solving and control to the Space Station Freedom Electrical Power System (SSF/EPS) testbed being developed and demonstrated at NASA Lewis Research Center. The objectives of the program are to establish artificial intelligence technology paths, to craft knowledge-based tools with advanced human-operator interfaces for power systems, and to interface and integrate knowledge-based systems with conventional controllers.

The Autonomous Power EXpert (APEX) portion of the APS program will integrate a knowledge-based fault diagnostic system and a power resource planner-scheduler. Then APEX will interface on-line with the SSF/EPS testbed and its Power Management Controller (PMC). The key tasks include establishing knowledge bases for system diagnostics, fault detection and isolation analysis, on-line information accessing through the PMC, enhanced data management, and multiple-level, object-oriented operator displays. The first prototype of the diagnostic expert system for fault detection and isolation has been developed.

This paper describes the knowledge bases and the rule-based model that has been developed for the Power Distribution Control Unit subsystem of the SSF/EPS testbed. A corresponding troubleshooting technique is also described.

INTRODUCTION

This paper is limited to the Autonomous Power EXpert (APEX) system, which is one of the major parts of the Autonomous Power System (APS) project under development at NASA Lewis Research Center, Cleveland, Ohio. The project is a joint effort of the Space Electronics, Power Technology, and Electrical Systems Divisions to develop and demonstrate the application of expert systems technology with a focus on the Space Station Freedom Electrical Power System (SSF/EPS) testbed.

APEX was designed as a high-level advisor for diagnosing faults in the Power Distribution Control Unit (PDCU), which is a major subsystem of the SSF/EPS testbed. The rule base and model of this subsystem were built based on both dynamic and static knowledge of the system configuration and the remote power-control devices (20 kHz Remote Power Isolator and Remote Power Controller, manufactured by Westinghouse Corp., Lima, Ohio). These remote power-control devices are used throughout the SSF/EPS testbed for controlling, protecting, and monitoring the power distribution traffic. An operational prototype of the expert system fault detection, isolation, and justification package was recently completed and tested successfully. Figure 1 shows the proposed integration of the APEX system with other subsystems (ref. 1).

Descriptions of the system knowledge bases, rule base and model, and troubleshooting technique are presented in the following sections.

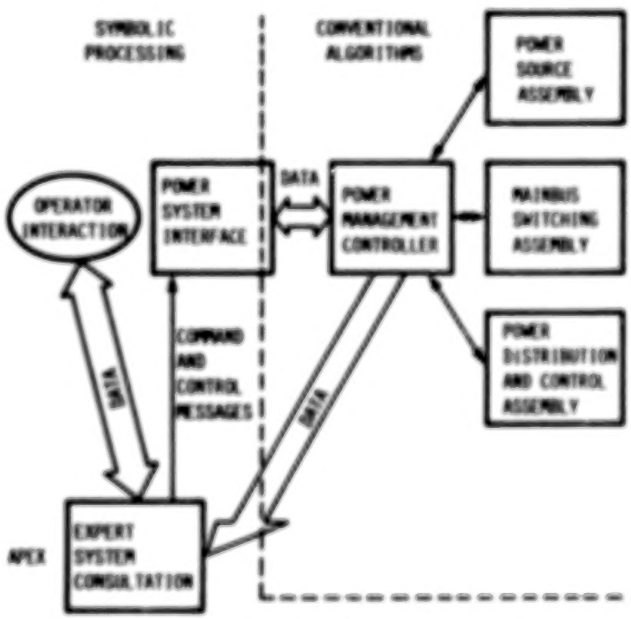


FIGURE 1. - AUTONOMOUS POWER SYSTEM INTEGRATION PLAN SHOWING AUTONOMOUS POWER EXPERT (APEX) SYSTEM.

SYSTEM KNOWLEDGE BASES

Design Activities

APEX is a knowledge-based system consisting of multiple knowledge bases that contain rules, facts, and graphic representations. To organize the data and to facilitate ease of maintenance and extendibility, we decided to use multiple knowledge bases. There are, therefore, four knowledge bases: one for the rules and facts, one for the user interface, and two containing graphic structures for human-interface panels.

The design of APEX is based on two primary requirements. The first is continuous monitoring of parametric data for incipient fault prediction. This type of monitoring requires retaining a history of data that can be checked for problems by trend analysis. The second requirement is multiple fault analysis of complicated failure modes. Conditions can occur that cause multiple failures in the system. These multiple failures can occur so quickly and generate so much data that only experts can analyze the resulting condition of the power system and take the appropriate actions to restore power. The objective for APEX is to capture the knowledge of the experts for these complicated situations and make the knowledge available to others who are responsible for the power system control.

Since the SSF/EPS is currently under design, not all of the diagnostic expert knowledge about the system is available. APEX will initially contain the current knowledge of the domain experts, but it will be extended to maturity as new knowledge is gained. By capturing the knowledge as it becomes available, APEX will retain and document new knowledge, as well as making it accessible to others.

In reference 2, expert systems are commonly developed on the basis of vague requirements. A general requirement to develop a system that does what the expert does is typical. For the APEX system, two specific requirements were identified before prototype development activity began: (1) incipient fault prediction and (2) multiple fault analysis. Specific functionality to meet these requirements was postponed until the prototype development phase of the project.

Architecture Selection

It was clear after several interview sessions with domain experts that the diagnostic procedural knowledge was best represented by a rule-based approach. The troubleshooting techniques that the experts use involve an "if something, then do" logic procedure to isolate faults in the system. An expert system development tool with the same logic capability was selected to help develop the prototype diagnostic system. The expert system tools being used for this development are the Knowledge Engineering Environments (KEE) software package by Intellicorp, Inc. (ref. 3), and the Texas Instruments (TI) Explorer II LX (trademark of Texas Instruments Corp.) computer workstation.

RULE BASE AND MODEL

Prototype Activities

We decided to model a correctly operating power system and to check primary data parameters to detect faults. As long as monitored data values were within tolerance according to the model, no rule firing was necessary. A combined forward-chaining (data driven) and backward-chaining (goal driven) approach was selected. Forward chaining was used for fault detection. Backward chaining with appropriate sets of rules for detecting faults was used for isolating probable causes of the faults. The rules were based on an up-to-date model of the PDCU. A frame-based inheritance network was used to represent the model.

A structured representation of all system components and their attributes was included in the model. Functional, behavioral, and physical properties of the components, as well as the interconnections between components, were represented. Upward and downward data trends were the primary bases for identifying incipient failure modes. Behavioral properties were used to describe current state information as well as expectations upon the occurrence of multiple faults.

A system for diagnosing fault conditions in a particular problem domain must know the limits of its capabilities and inform a human user of any problems. Many rules have been developed to detect when APEX is reasoning with inconsistent information that is not valid within its domain. A simple example of this phenomena is power output from a device that is "off" according to status information. Such information is often difficult to encapsulate in any type of program and can cause inappropriate rule firing leading to incorrect conclusions. Much effort was made to avoid any situations that could mislead an operator.

The knowledge base can be extended to include additional rules and diagnostic procedures as new knowledge becomes available.

Parametric Data Simulation

Parametric data available from the electrical power system testbed instrumentation include analog and digital data values. The analog values are power, voltage, current, and phase angle at various test points. The digital values provide status information. For prototyping purposes, a software simulator was developed to provide data for testing the expert system. Simulated data can be conveniently generated via a mouse-activated simulator panel as shown on figure 2.

Features of the simulator include initialization of the system, fault injection, and automatic calculation of parametric data for all test points of the PDCU. Initialization sets the system to a steady-state, no-fault configuration. Fault injection provides a method to generate fault conditions. Test point data are calculated automatically by engineering algorithms that simulate the behavior of the PDCU circuitry.

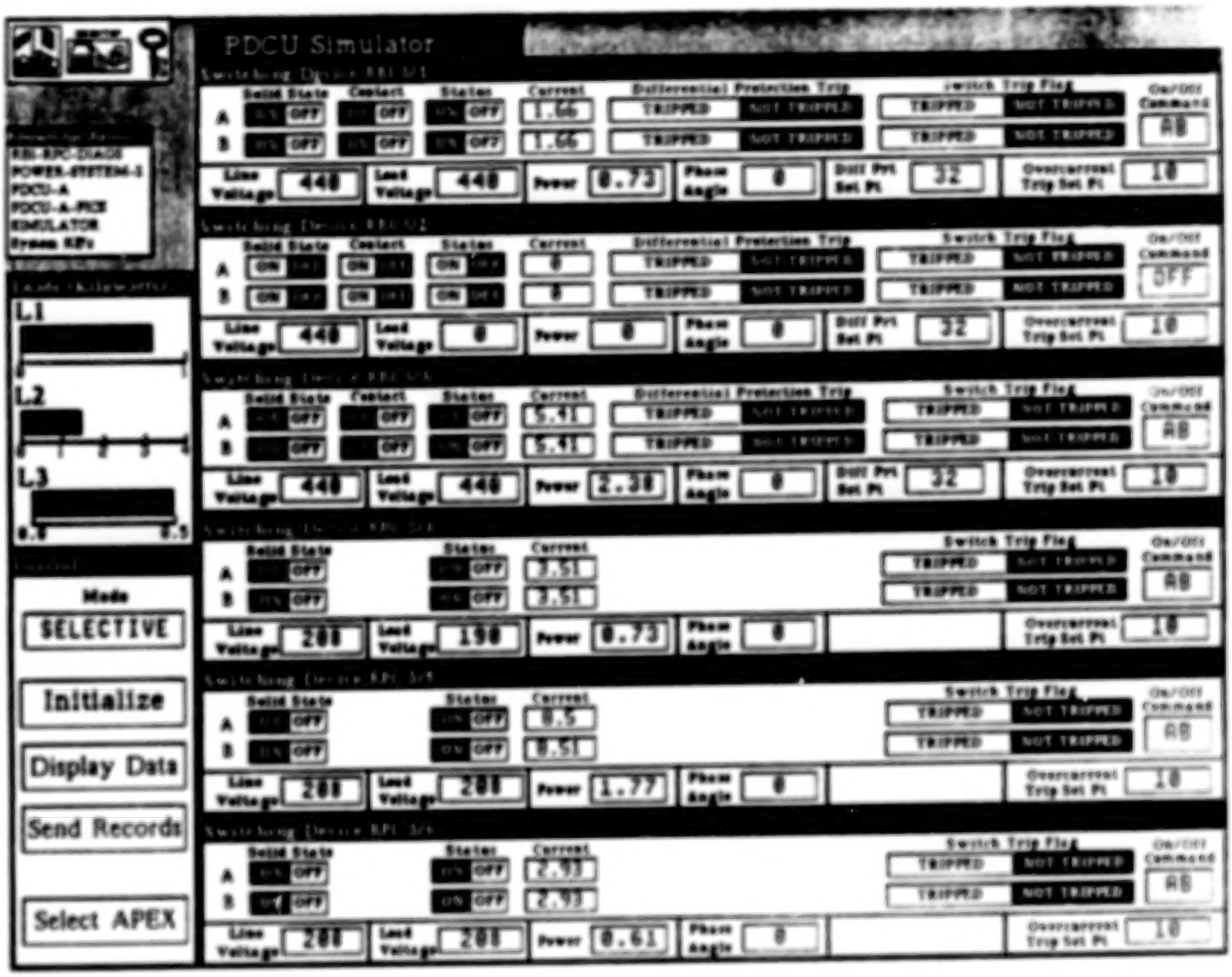


FIGURE 2. - DISPLAY OF AUTONOMOUS POWER EXPERT (APEX) POWER DISTRIBUTION CONTROL UNIT SIMULATOR.

Simulated data values are written to a memory area that is shared by the expert system via a blackboard communication interface. During the next phase of the project, the PMC will replace the simulator. After APEX is interfaced with the PMC for further development, actual data will be read and communicated to the same blackboard interface.

TROUBLESHOOTING TECHNIQUE

All identifiable faults in the PDCU can be detected and isolated. These faults include failures in individual or combinations of power sources, remote bus isolators, remote power controllers, transformers, transmission lines, and electrical loads. The troubleshooting technique has three main features: fault detection, isolation of probable causes, and justification for the probable causes.

Within the framework of the diagnostic knowledge base, the rules are organized into separate frames for detection and isolation. Forward chaining, driven by changing data in the blackboard communications memory area detects faults within the system. Faults can either be caused by parametric data that falls outside of the desired model, or a trend in the data that indicates an incipient failure. Backward chaining is used to reach conclusions concerning fault conditions.

Justification retrieves the reasoning path used to reach conclusions and displays natural language explanations of the reasoning. The explanations are used to justify conclusions to the operator. The natural language interface was developed in LISP (LIST Processing language) with schema functions linked to the rules in the knowledge base. Rules that fire during fault isolation are retrieved in the order that they fired and are linked to natural language explanations by the schema functions.

The natural language explanations appear on several of the operator panel displays. Some of the explanations have further justifications that provide deeper levels of explanation. Complete explanations of the reasoning used to reach conclusions is provided through the schema-based natural language interface.

Human-interface panels provide three levels of information accessing to the system. The highest level of access is a two-panel display that consists of an operator panel and a mouse-activated panel showing the power system overall block diagram. Figure 3 is a hard copy of the APEX top level display. The operator panel provides operator controls for activating blackboard monitoring, detecting faults, isolating causes, requesting justification, and printing out desired information. Mousing on subsystem displays of the PDCU overall block diagram panel provides access to lower-level subsystem block diagrams and circuit schematics as shown in Figures 4 and 5, respectively.

All of the displays automatically highlight any problem areas and specific fault locations. These highlighted blocks provide information to the operator and display current data values from the blackboard memory area.

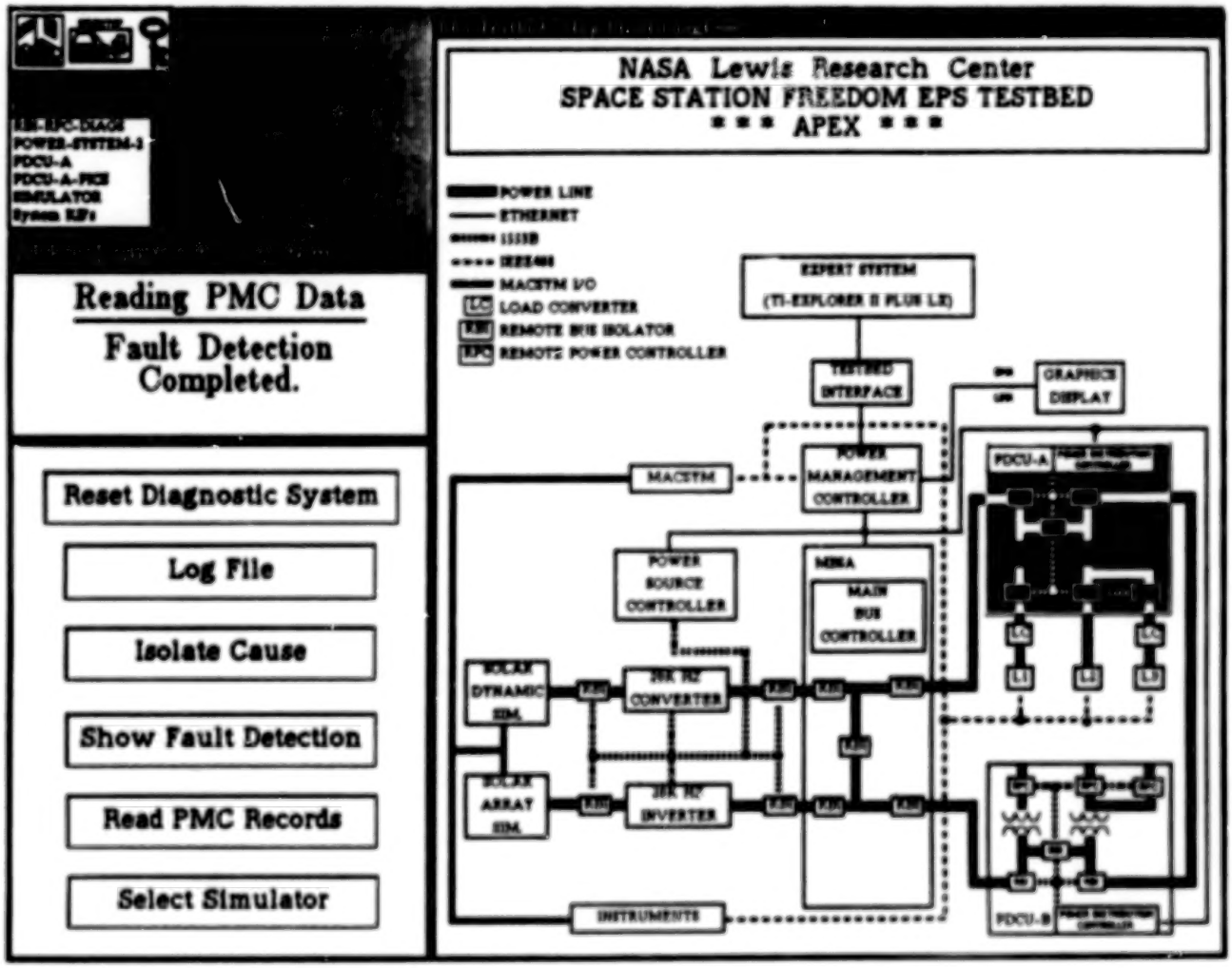


FIGURE 3. - AUTONOMOUS POWER EXPERT (APEX) TOP LEVEL DISPLAY SHOWING OPERATOR PANEL AND ELECTRICAL POWER SYSTEM (EPS) TESTBED.

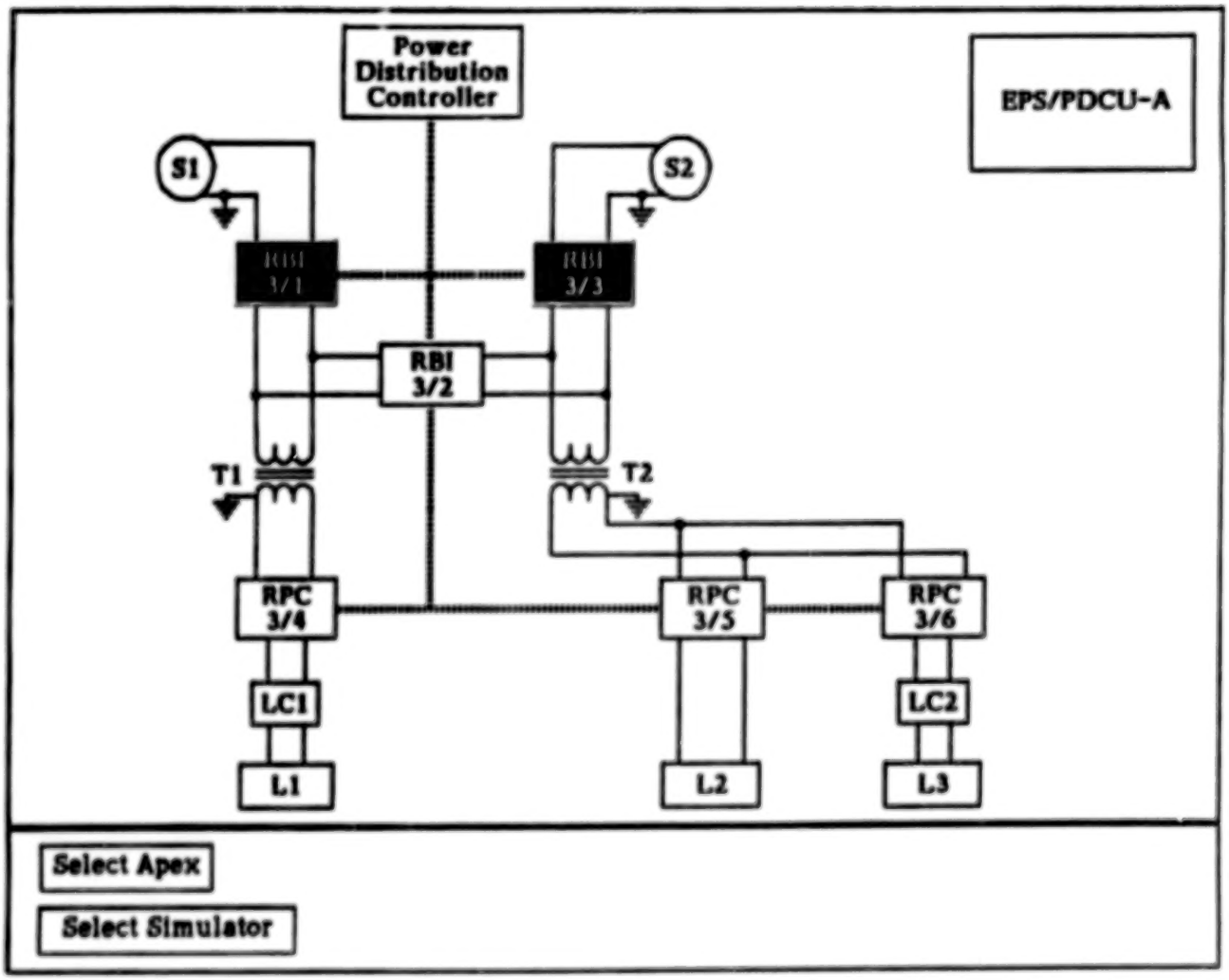


FIGURE 4. - POWER DISTRIBUTION AND CONTROL UNIT SCHEMATIC DIAGRAM SHOWING REMOTE POWER CONTROLLERS (RPC), REMOTE BUS ISOLATORS (RBI), LOAD CONVERTERS (LC), LOADS (L), 20-kHz POWER SOURCES (S), AND TRANSFORMERS (T).

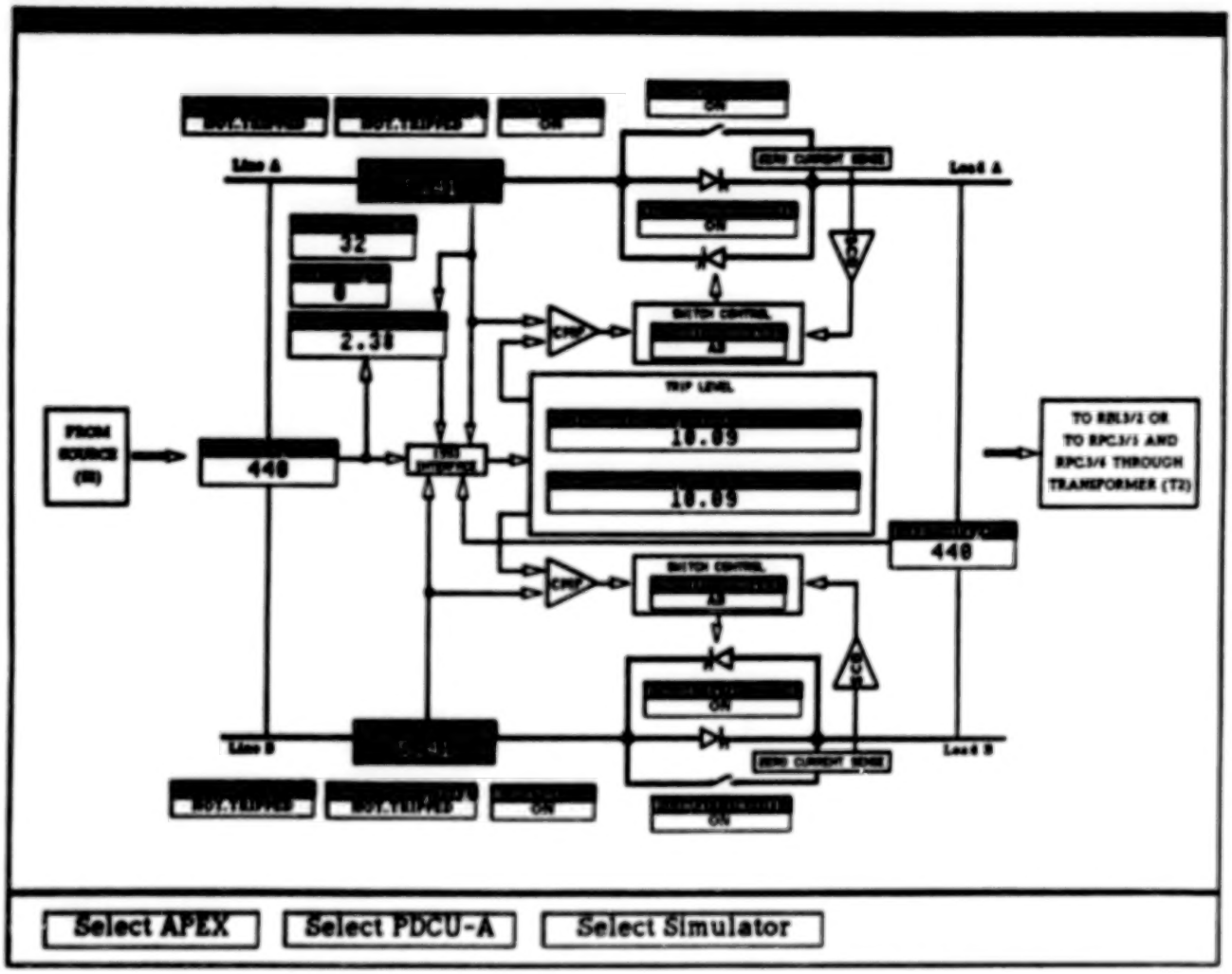


FIGURE 5. - SIMPLIFIED CIRCUIT DIAGRAM OF REMOTE BUS ISOLATOR SHOWING DATA VALUES.

CONCLUSIONS

The Autonomous Power EXpert (APEX) system is fully mouse activated and menu driven for quick and easy operation. Its knowledge bases are well constructed for fast, yet sufficient on-line information. The troubleshooting technique and rules are customized for quick detection and isolation of incipient, and multiple or single faults. Natural language explanations are implemented to show the reasoning applied during the process of isolating faults. These explanations benefit the operator of the testbed by justifying the accuracy of fault isolation conclusions and are reviewed by domain experts to verify proper operation of the APEX system. APEX is also flexible to accommodate changes in the testbed hardware configuration.

Integration of APEX with the PMC and the power resource planner-scheduler is the next logical step. Future expansion of this work could lead to direct applications and delivery of expert system technology to the full-scale operation of the Space Station Freedom Electrical Power System testbed at NASA Lewis Research Center.

REFERENCES

1. Dolce, J.L., "An Integrated Approach to Space Station Power System Autonomous Control," 22nd IECEC, Vol. 1, AIAA, New York, NY, 1987, pp. 499-508.

2. Culbert, C., Riley, G., and Savely, R.T., "Approaches to the Verification of Rule-Based Expert Systems," FIRST ANNUAL WORKSHOP ON SPACE OPERATIONS AUTOMATION AND ROBOTICS, SOAR '87, NASA CP-2491, National Aeronautics and Space Administration, Washington, DC, 1987, pp. 191-196.

3. KEE USER'S GUIDE, Version 3.1, Intellicorp Inc., Mountain View, CA, May 1988.

An Expert System to Advise Astronauts During Experiments: The Protocol Manager Module

Guido Haymann-Haber*,
Silvano P. Colombano**,
Nicolas Groleau****,
Don Rosenthal**,
Peter Szolovits***,
Laurence R. Young****

* Knowledge Systems Laboratory, Stanford University, Palo Alto, CA 94304, U.S.A.
** Artificial Intelligence Research Branch, NASA-Ames Res. Ctr. Mail Stop 244-17, Moffett Field, CA 94035, U.S.A.
*** Laboratory for Computer Science, Massachusetts Institute of Technology. Cambridge, MA 02139, U.S.A.
**** Man-Vehicle Laboratory, Massachusetts Institute of Technology. Cambridge, MA 02139, U.S.A.

Abstract

Perhaps the scarcest resource for manned flight experiments - on Spacelab or on the Space Station Freedom - will continue to be crew time. To maximize the efficiency of the crew, and to make use of their abilities to work as scientist collaborators as well as equipment operators, normally requires more training in a wide variety of disciplines than is practical. The successful application of on-board expert systems, as envisioned by the "Principal Investigator in a Box" program, should alleviate the training bottleneck and provide the astronaut with the guidance and coaching needed to permit him or her to operate an experiment according to the desires and knowledge of the PI, despite changes in conditions. This report covers the Protocol Manager module of the system. The Protocol Manager receives experiment data that has been summarized and categorized by the other modules. The Protocol Manager acts on the data in real-time, by employing expert system techniques. Its recommendations are based on heuristics provided by the Principal Investigator in charge of the experiment. This prototype has been developed on a Macintosh II by employing CLIPS, a forward-chaining rule-based system, and HyperCard as an object-oriented user interface builder.

Introduction

There is an ongoing interest in understanding the phenomenon of space motion-sickness. This is a pervasive problem that, aside from producing discomfort to the astronauts, has at times significantly reduced their effectiveness during space missions.

In trying to shed light on the physiological principles that underlie this phenomenon, several experiments have been conducted both during space missions and on the ground. While the Principal Investigator (PI) in charge of the experiments can supervise the tests conducted on the ground, the astronauts have to act both as subjects and operators of the experiments while in space. Without the supervision of the PI, the data collected during past space trials has not been as complete and useful as desired. The crews have been unable to deal adequately with unexpected events, and make the necessary adjustments to the experiments. The PI, on the ground, cannot always be counted on for directions as he or she may have neither the accurate real-time data needed to make a decision, nor be able to communicate with the astronauts. The astronauts are generally unprepared to make these decisions, as the numerous activities they are expected to perform during the mission only allow them to have a basic idea of the nature and objectives of the experiments.

The experiments are conducted throughout the space mission during several sessions of approximately one hour. The schedule and content (the Protocol) of the sessions is prepared before the mission is launched. Once the mission is launched however, several events can affect that plan and necessitate changes to either the schedule or the Protocol of the sessions. These events include equipment malfunctions, running short of time, finding interesting data that the PI may want to investigate, or having a sick or otherwise unavailable astronaut.

We are proposing to provide a computer-based advice system that will help the astronauts to cope with these problems by assisting them in the detection of out-of-bounds conditions, in analyzing these conditions, and in suggesting alternative courses of action. With this system, the astronauts will have available some of the expertise of the Principal Investigator. The system is thus appropriately called *PI-in-a-Box*, or ▣ for short.

The next set of experiments will be conducted during the Space

Laboratory Mission (SLS-1) of the Space Shuttle, currently scheduled to be launched in 1990. A prototype of some of the modules of the system will be tested in flight in 1992 and on the ground during 1990.

The key success factors for the system are:

- The astronauts must perceive that the system provides obvious advantages for them.
- The use of the system must be optional. The experiment must be able to proceed independently of whether the system is being used or not.
- The implementation must be as hardware-independent as possible since the alternatives that eventually will be available during the space mission are unknown.

The subject of this report, is one module of *PI-in-a-Box*, the Protocol Manager. Reference 3 provides detailed descriptions of the design and implementation of this system. Early prototypes of other modules are described in reference 1.

Domain Description

This section provides an overall description of the domain addressed by this particular system. References 5 and 6 provide detailed discussions of the underlying physiological issues.

The experiment addressed by this system is called the Rotating Dome Experiment. Its purpose is to study the interaction of several spatial orientation senses during and following adaptation to weightlessness. Normally all the senses (visual, vestibular, proprioceptive, tactile) act in harmony during voluntary head movements. In orbit, however, the organs of the inner ear, no longer produce signals which the brain can use to deduce the angular orientation of the head with respect to the vertical - and of course the vertical itself ceases to have any real significance. Nevertheless, the brain still searches for a reference system, within which it can place external (seen) and body position measurements. Visual cues, both static and dynamic, as well as localized tactile cues, may become increasingly important in signalling spatial orientation.

A better understanding of the level of brain adaptation to altered gravio-inertial forces may help to explain and possibly alleviate the symptoms of space motion sickness, which are thought to be related to sensory-motor conflict concerning spatial orientation.

During the Dome Experiment, the subject's field of vision is filled by a dome. This dome rotates at various speeds and directions, while several measurements are being taken. The dome operation normally entails a one hour experiment with two astronauts - alternating as subject and operator. This period, referred to as an Experiment Session, is repeated several times throughout the space mission. In addition, the experiment is also performed on the ground during the days preceding the flight in order to get baseline data, and immediately following the mission in order to study the readaptation to the Earth's gravity.

An Experiment Session starts with un-stowing, setting-up and testing the equipment, and preparing the subjects.

The experiment is paced by a dedicated computer, the Experiment Control and Data Systems (ECDS), which generates instructions, starts and stops the dome rotation according to pre-programmed sequences, acquires, digitizes and transmits data, and permits routing of analog test signals for hardware testing and for calibration.

Each subject will normally take part in three runs under different conditions:

- The free float condition has the subject restrained by a bite-board and his or her right hand on a joy stick.
- The neck twist condition is like the previous one, except that the subject starts each dome trial by tilting his or her neck.
- The bungee condition has the subject held down to a foot restraining grid plate by stretched elastic bungee cords.

Each run lasts about 3 minutes. After the experiment, the equipment is deactivated and stored.

During the course of an experiment several types of data are recorded. These include:

A joy stick signal from a potentiometer adjusted by the subject. The subject uses it to indicate the strength of his or her visually induced rotation rate relative to the speed of the dome. Full deflection of the potentiometer clockwise, for example, would indicate that the subject felt that he or she was rotating to the

right and that the dome (which was actually turning counter-clockwise) was apparently stationary in space.

A bite-board measures neck torque by means of strain gauges attached to the support. It measures the tendency of a subject to straighten out his or her head to the upright when sensing that he or she is falling.

Neck muscle EMG from the right and left sides are also indicators of the initiation of righting reflexes to straighten the head.

The astronauts perform the experiment by following a checklist with detailed step by step instructions. This checklist is prepared by the PI before the space mission. Unfortunately, the astronauts often must deviate from this pre-defined protocol due to a variety of circumstances such as:

- The experiment is running late. This could, among other things, be due to a late start or delays in performing some of the steps of the experiment. Since the ending time of the session is strictly enforced, some parts of the experiment may have to be eliminated.

- There are equipment problems. A piece of equipment may have failed, possibly degrading the quality of the data. A decision has to be made as to whether to continue the experiment with degraded data or to spend valuable session time trying to troubleshoot and fix the problem.

There are some additional circumstances in which a change in the protocol might be desirable, and that are very difficult for the astronauts to perceive, such as:

- The data being collected from the subject is "interesting." It might be desirable to perform some additional runs on that subject.

- The subject is providing "erratic" data that is not very useful. It might be desirable to concentrate on the other subject.

Communication channels between the spaceship and the ground may not be available for experiment use during a session. Consequently, the PI generally does not have real-time access to the data or the astronauts. As a matter of fact, even if this were possible, he may not have enough time to analyze the data and make a recommendation. In previous missions, where similar experiments were conducted, a significant amount of potentially useful data was never collected due to circumstances such as those mentioned above.

The PI-in-a-Box System

The *PI-in-a-Box* system has been divided into several relatively independent modules. It is centered around the Protocol Manager, which is the subject of this report. The Protocol Manager monitors and suggests proper actions during an experiment session. In addition, the system includes the following modules:

- A Data Quality Monitor, that monitors the output from the data collection system and pinpoints suspect signals.

- A Diagnosis and Troubleshooting System, that assists in the diagnosis and repair of equipment.

- An Interesting Data Filter, that detects interesting or unexpected patterns in the measurements.

- An Experiment Suggester, that comes up with additional tests that might be run if spare time is available.

- A Scheduler, that does long term scheduling of experiment sessions throughout the mission (not to be confused with the scheduler of an operating system).

- An Experiment Controller (ECDS), that controls the operation of the dome.

- A Signal Processing module that picks up the signal from the Experiment Controller (ECDS).

These modules interact with each other and with the Astronauts and the PI. A diagram of their interactions is shown in figure 1. It is important to note that the arrows represent the flow of control, not data. The latter may move directly between any two modules as needed.

The system must operate in real-time. It needs an Executive to schedule the execution of the appropriate module, since different events may compete for computing resources. The aim is to make the system as independent as possible from the hardware or operating system configurations on which it may be implemented. Consequently, few assumptions have been be made about the

architecture of the Executive in the design of each of the modules, and standardized interfaces have been defined.

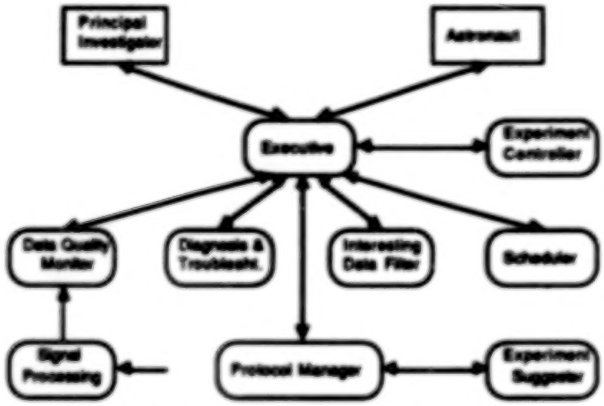


Fig. 1: Control flow between the modules

Because of the complex user interfaces required by *PI-in-a-Box*, the development of this project has been done on the Apple Macintosh II. Since the computers used during the mission have to be space-qualified, which entails a series of rather lengthy and expensive tests, there is no guarantee that a Macintosh will be available during the mission. Consequently, the system must be developed with the ability to be ported in relatively short order to some of the available hardware platforms.

The Protocol Manager uses a combination of a rule-based system in order to do the "reasoning," and a fast application builder in order to build the user interface.

For the "reasoning" part, CLIPS (*C Language Integrated Production System*) was used (ref. 2). CLIPS is a forward-chaining rule-based system that combines some of the features of the expert system shells of OPS5 and ART. CLIPS was developed and is supported by NASA. It was chosen because currently there are versions for both the Macintosh and 80x86-based computers, making it easy to port between any of the environments. In addition, the CLIPS source code, written in C, is available and can be customized in order to handle specific needs. Finally, CLIPS is a fairly simple and yet powerful language.

The user interface was built using HyperCard (ref. 4). Although it has several important restrictions, HyperCard provides a good environment to build complex user interfaces.

The general philosophy has been to shift as much as possible of the "reasoning" processes to CLIPS. CLIPS is easy to port to other environments, while HyperCard probably will be replaced in the final version.

Using the Protocol Manager

Most of the visible activity of the Protocol Manager is centered around the Session Manager. A typical screen is shown in figure 2.

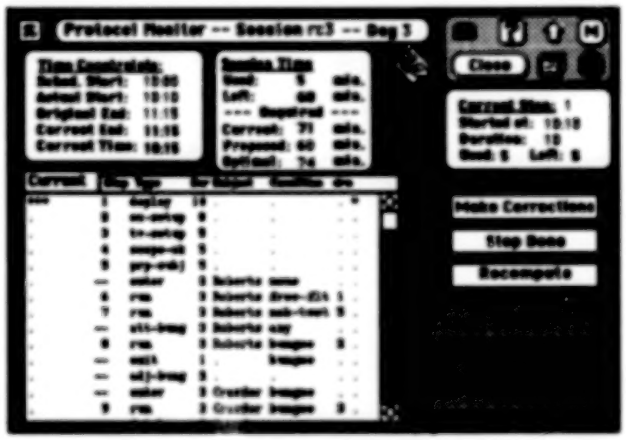


Fig. 2: Typical Protocol Manager screen

In this example, the astronaut has just started a new session. The top box indicates that this is the third day in the mission and that this session is code-named "rc3."

The Time Constraints box shows that this session was scheduled to start at 10:00. However, the astronaut has indicated that it actually started at 10:10. The scheduled ending time remains unchanged at 11:15. The current time is 10:15, that is, 5 minutes into the session.

The Session Time box indicates that 5 minutes of the current session have been used and that 60 are left. Below, the Session Manager reports that 71 minutes would be required in order to complete the protocol proposed by the PI. This is 11 minutes past the scheduled ending time. However, the Session Manager indicates that it has an alternative protocol to propose. It would fit within the time that is available, taking exactly 60 minutes. There also is an "optimal" protocol that includes everything that the Protocol Suggester would like to try. It takes 74 minutes.

The protocol steps themselves are displayed in the window below. The stars (***) identify the step that the Protocol Manager believes that is currently being performed. For each step there is an associated step number, a description (Type), an expected duration (in minutes), and if applicable, the name of the

subject, the condition and a dome run number (drn). The astronauts for this session are Roberts and Crawford.

The Current Step box on the right-hand side indicates some basic data about the current step.

The system includes on-line help and the possibility to enter user comments. The protocol display may also be used as a checklist. By clicking on the appropriate step, the detailed instructions that need to be performed are displayed.

The astronaut may examine the alternative protocol that has been proposed by clicking on the **Current** heading, and selecting the **Proposed** protocol, as shown in figure 3.

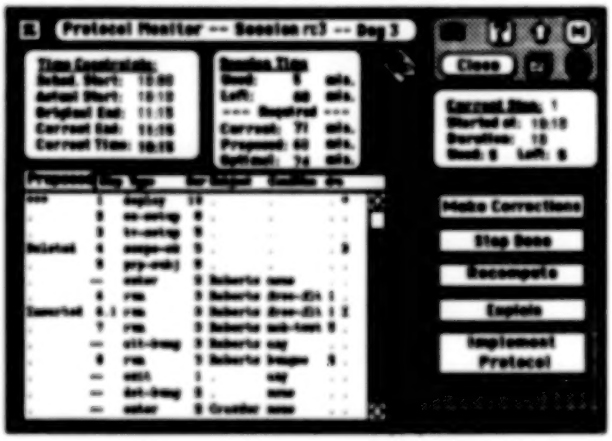


Fig 3: Alternative protocol proposed at the start of the Session

One of the changes that is being recommended is to delete the *scope check* step. In addition the Protocol Suggester recommends to insert a second *free float* run (step 6.1). The astronaut may click on step 6.1 in order to get an explanation for this recommendation, as shown in figure 4.



Fig. 4: Explanation for recommended step 6.1

The explanation indicates that interesting data was found on Roberts during the last session and was not investigated. This triggers a request for a double run of the *free float* condition, with an associated force of 2 (see next section).

If the astronaut decided to adopt the proposed protocol, he or she just needs to click on the **Implement Protocol** button.

The Protocol Manager is designed to operate with a minimum of input from the astronauts. It will make the proper assumptions about the state of the experiment session based on the signals it gets from the other modules. The astronauts may modify any of these assumptions.

For instance, in the session displayed above, Roberts is scheduled to be the first subject. If after completing the preparations for the experiment, the astronauts decide to reverse this order, they may indicate this to the Protocol Manager, which will suggest a modified protocol. The protocol, as accepted by the astronaut, is shown in figure 5.

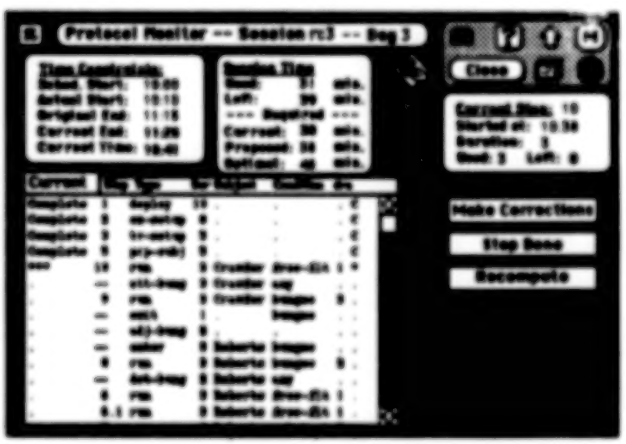


Fig. 5: Modified protocol with subjects switched

In the new protocol, the currently scheduled runs for Roberts are moved and re-inserted in a different order after the bungee run for Crawford. The run sequence for Roberts has been altered in order to take advantage of the fact that the bungee will be attached when Crawford exits the dome. This saves time.

Notice that a 5 minute time extension has been granted since the start of the experiment (the ending time is now 11:20). Nevertheless, step 11, the *neck twist* run for Crawford has been cut anyway in favor of a double *free float* run for Roberts. If the

astronaut were to request an explanation, the contents of figure 6 would be shown.

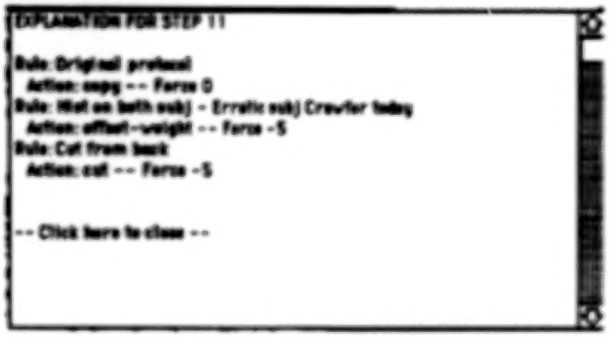


Fig. 6: Explanation for the elimination of step 11

Each step has an associated weight, which is the result of recommendations of varying forces. This concept is used in order to select the most important steps. It is explained in more detail in the next section.

The explanation depicted in figure 6 says that the *hunger* step was contained in the original protocol and is thus included with force 0. However, during the last run, Crawford's measurements were erratic, which prompted a reduction in the weight of his run by 5 units of force. Consequently, given that there was not enough time, the step was cut.

The Protocol Manager contains several other facilities that include the ability to study the history of previous sessions, undo actions, modify a series of decision parameters, and so forth.

Protocol Manager Architecture

The diagram of figure 7 presents an overall view of the environment in which the Protocol Manager operates:

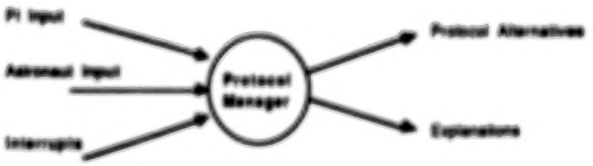


Fig. 7: Overall environment of the Protocol Manager

The PI and the Astronauts can interact with the Protocol Manager by entering, updating or requesting information. In addition, the Protocol Manager reacts to certain messages sent by the other modules of the system. These messages are referred to as interrupts. The Protocol Manager, in turn, provides protocol alternatives and explanations for its recommendations to its users.

The Protocol Manager has been broken down into two components. A Session Manager that handles all the interactive work, and a Protocol Suggester, that operates invisibly underneath the Session Manager and provides it with new protocol alternatives upon request. The Session Manager and the Protocol Suggester interact by passing data to each other as illustrated in figure 8. The arrows represent the data flow between the two components.

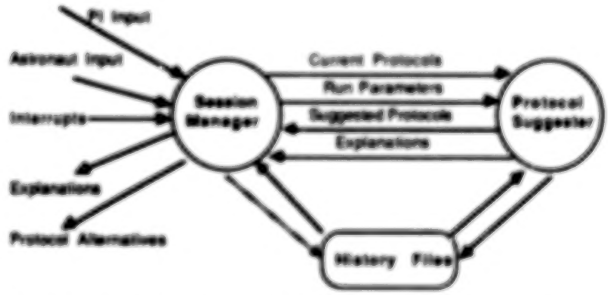


Fig. 8: Data flow between the Session Manager and the Protocol Suggester

The Session Manager periodically requests new protocol alternatives from the Protocol Suggester. As illustrated in the diagram, the modules communicate by sending data directly to each other, and through history files where all relevant events that have occurred during the mission are saved. Note the shaded link from the Session Manager to the History Files. This is a "development link" used during the testing phase of the system in order to set up different scenarios.

There are three main motivations that favor this division of the Protocol Manager:

- The Protocol Manager must be "aware" of time. It must be available upon request and it must take actions that are triggered at certain time intervals. Most rule-based systems (and CLIPS is no exception) operate on a run-cycle concept; they read input values, fire the appropriate rules, and produce a result. During this period, conditions are assumed to stay constant, that is, time is static. This is clearly not an adequate environment for the above system. By virtue of this division, the Session Manager acts as a supervisor that decides when it is necessary and possible to update the suggested protocol.

- The Protocol Manager must receive input from the other modules and the users in real time. It must provide reasonably fast responses to what in general are simple requests (such as a request to display the instructions for a particular step). For the reasons stated above, CLIPS is not appropriate for this type of operation.

- Most of the "intelligence" of the system lies in the process of suggesting a protocol, and a significant effort must be put into its development and maintenance. It is very desirable that the implementation of this process be as independent as possible from the hardware or system software. By making the Protocol Suggester a "black box" with a minimum of assumptions about the operating environment, the modifications that may result from system configuration changes are limited mosily to the Session Manager.

The **Session Manager** then, handles the interaction with the users and the interface with the other components of *PI-in-a-Box*. In essence, the Session Manager provides the astronauts with an intelligent checklist that displays the progress of the session and provides alternatives for action based on the output of the Protocol Suggester and its knowledge about the environment.

The **Protocol Suggester** is subordinate to the Session Manager, and provides it with new protocol alternatives upon request. In broad terms, the process of suggesting a protocol consists of three stages:

I Proposing a series of actions to take given the state of the current protocol and knowledge about the past history of the current and previous sessions.

II Generating all the steps that should be executed in order to comply with the proposed actions.

III Assembling the "best possible" protocol from those steps, that complies with the time constraints of the current session.

This process model represents a key decision in the design of the Protocol Suggester. During the conversations with the PI it became apparent that there were two sets of heuristics: heuristics to decide which steps to include in the protocol, and heuristics to decide in which sequence to perform the steps.

Since generally there are more steps that are desirable to perform than there is time to actually perform them, a complex interaction ensues between all the different heuristics in order to decide which particular steps to perform in any given context. There is clearly the potential for an explosive growth of the number of combinations. This could make the system unmanagable, un-maintainable, and slow.

The solution adopted was to introduce the concept of step weight. Each step has a weight associated with it. This weight reflects the importance of the step, the higher the weight, the more desirable it is to perform the step. Through this artifact, the problem is broken down into two independent parts: determining which steps to perform, and choosing and ordering the steps with the highest weights that fit within the allotted time. The former is performed by stages I and II, while the latter is done during stage III.

There may be one or more heuristics which favor the inclusion of a particular step. These heuristics are expressed in stage I by proposing actions. Each of these actions has an associated force. The forces of all the actions proposing a particular step are combined in order to produce the weight of that step. The current heuristic is to simply make the weight of the step equal to the highest force of all the actions that propose that step. This is done as part of stage II.

While the use of weights is a completely arbitrary solution, it has provided a surprising flexibility in adjusting the actions for each scenario. The main disadvantage of this approach is that in the explanations for the inclusion or exclusion of a step, the causal chain that leads to the result is somewhat blurred.

The main advantage of the "weight" approach is, of course, the avoidance of a combinatorial explosion of rules. Adding a new rule is mostly a linear process, with few, if any, side effects to the other rules. Another advantage is that the system is more robust; if a particular combination of circumstances has not been contemplated, the Protocol Suggester will provide a reasonable answer, even though it may not be the best.

After each invocation, the Protocol Suggester returns the following information to the Session Manager:

- An optimal protocol, that is, a protocol that includes all the steps that the Protocol Suggester would like to see executed, without regard to the time it would take to perform them. In other words, all the steps generated during stages I and II are included, regardless of their weight.

- A proposed protocol, that is, a protocol that fits within the time currently allotted to the session. This protocol is a subset of the optimum protocol. However, the steps may be in a different sequence.

- A set of explanations, justifying the inclusion or exclusion of each step from the protocol.

Conclusions

This prototype system has shown that it is possible to design a fairly sophisticated experiment protocol manager for use in space. Furthermore, it is possible to do it with relatively unsophisticated hardware and software.

The main challenge in the design of such a system is to conceive a suitable software platform that provides a good paradigm for future growth and maintenance of the system. We believe that this has been achieved.

Important challenges lay ahead. These include the ability to make all the modules work together in real-time, and producing a good user interface.

Acknowledgements

This project was made possible by NASA grant NCC 2-570 and RTOP 506-47-11 for "Crew Station Design," respectively from the AI Research Brand and the Human Factors Division at NASA-Ames.

The Stanford University Knowledge Systems Laboratory and Apple Corporation have provided generous support to this project.

We would also like to acknowledge the contribution of Dr. Irving Statler and the members of the Human Factors Division. Dr. Statler's ideas and encouragement for serious user interface work will hopefully come to fruition in a later version of *PI-in-a-Box*.

References

1. Colombano, S., Young, L.R., Wogrin, N., Rosenthal, D. "PI-in-a-Box: Intelligent Onboard Assistance for Spaceborne Experiments in Vestibular Physiology." *Proceedings of the IV Conference on AI in Space Applications*, November 1988.

2. *CLIPS Reference Manual*. Version 4.2. Artificial Intelligence Section, Lyndon B. Johnson Space Center, April 1988. Program and manual available from "CLIPS Users Help Desk./M30. Computer Sciences Corporation. 16511 Space Center Boulevard. Houston, TX 77058. (713) 280-2233.

3. Haymann-Haber, G. *PI in a Box: The Design of an "Expert" Protocol Manager*, Knowledge Systems Laboratory Report 89-55. Stanford University, 1989.

4. *Apple HyperCard Script Language Guide: The HyperTalk Language*. Addison-Wesley, 1988.

5. Young, L.R., Shelhamer, M., Modestino, S.A. "M.I.T./Canadian Vestibular Experiments on the Spacelab-1 Mission: 2. Visual Vestibular Interaction in Weightlessness," *Experimental Brain Research 64*: 299-307, 1986.

6. Young, L.R., Oman, C.M., Watt, D.G.D., Money, K.E., Lichtenberg, B.K., Kenyon, R.V., Arrott, A.P. "M.I.T./Canadian Vestibular Experiments on the Spacelab-1 Mission: 1. Sensory Adaptation to Weightlessness and Readaptation to One-g: an Overview," *Experimental Brain Research 64*: 291-298, 1986.

The Environmental Control and Life Support System Advanced Automation Project

Phase 1 - Application Evaluation

Brandon S. Dewberry
NASA / Marshall Space Flight Center
System Software Branch (EB42)
MSFC, AL 35812

Abstract

The Environmental Control and Life Support System (ECLSS) is a Freedom Station distributed system with inherent applicability to advanced automation primarily due to the comparatively large reaction times of its subsystem processes. This allows longer contemplation times in which to form a more intelligent control strategy and to detect or prevent faults.

The objective of the ECLSS Advanced Automation Project is to reduce the flight and ground manpower needed to support the initial and evolutionary ECLS system. Our approach is to search out and make apparent those processes in the baseline system which are in need of more automatic control and fault detection strategies, to influence the ECLSS design by suggesting software hooks and hardware scars which will allow easy adaptation to advanced algorithms, and to develop complex software prototypes which fit into the ECLSS software architecture and will be shown in an ECLSS hardware testbed to increase the autonomy of the system.

This report covers the preliminary investigation and evaluation process, aimed at searching the ECLSS for candidate functions for automation and providing a software hooks and hardware scars analysis. This analysis will show changes needed in the baselined system for easy accommodation of knowledge-based or other complex implementations which, when integrated in flight or ground sustaining engineering architectures, will produce a more autonomous and fault tolerant Environmental Control and Life Support System.

Domain Overview

For development purposes, the ECLSS has been divided into six subsystems:

1. Temperature and Humidity Control (THC)
   - Condensate Removal
2. Water Recovery Management (WRM)
   - Potable Water Recovery
   - Urine Water Reclamation Pre-treatment
   - Hygiene Water Recovery
3. Air Revitalization (AR)
   - Carbon Dioxide Removal
   - Carbon Dioxide Reduction
   - Oxygen Generation
   - Trace Contaminant Control
4. Atmosphere Control and Supply (ACS)
5. Waste Management (WM)
6. Fire Detection and Suppression (FDS)

The first three (THC, WRM, and AR) have components which interact to cycle water and air for the crew as illustrated in figure 1. The air revitalization and water recovery management functions introduce new technology to NASA programs; extensive analysis and pre-flight testing is being performed at MSFC to insure proper operations and procedures necessary to control these cycles. One control objective is to isolate the operations of the subsystems such that these subsystems can be operated independently without analysis of their overall fluid loops. NASA ECLSS hardware and control engineers have expressed concern that the control and fault detection methods planned for the baselined ECLSS do not take into account the overall air and water generation loop interactions. Stable control of these loops will be difficult if not impossible to achieve by simply adjusting knobs in the test environment, scheduling setpoint adjustments of the process control components.

The control and software architecture to manage the ECLSS is hierarchically, physically, and functionally divided as illustrated in figure 2. The lowest level of control is that of real-time process control systems which perform the actual chemical transformations. The next higher level is the ECLSS element supervisor, which monitors and maintains those process controllers physically contained inside a Freedom Station Element, such as the Laboratory. The highest level of ECLSS control is the ECLSS Station Manager. It performs those functions which require knowledge from across element boundaries. It also interacts with the Operations Management Application (OMA) to perform distributed system resource allocation and consumption management with the rest of the Station distributed system managers, such as the Electrical Power System (EPS) manager.

In Phase 1 of the ECLSS Advanced Automation Project we are in the process of analyzing both the functional interactions of the regenerative processes and the software control architecture in search of candidate applications for advanced automation.





Air Revitalization Subsystem
Hygiene Water Recovery Subsystem
Potable Water Recovery Subsystem
Temperature and Humidity Control Subsystem

Figure 1 - Regenerative ECLSS Functional Interfaces

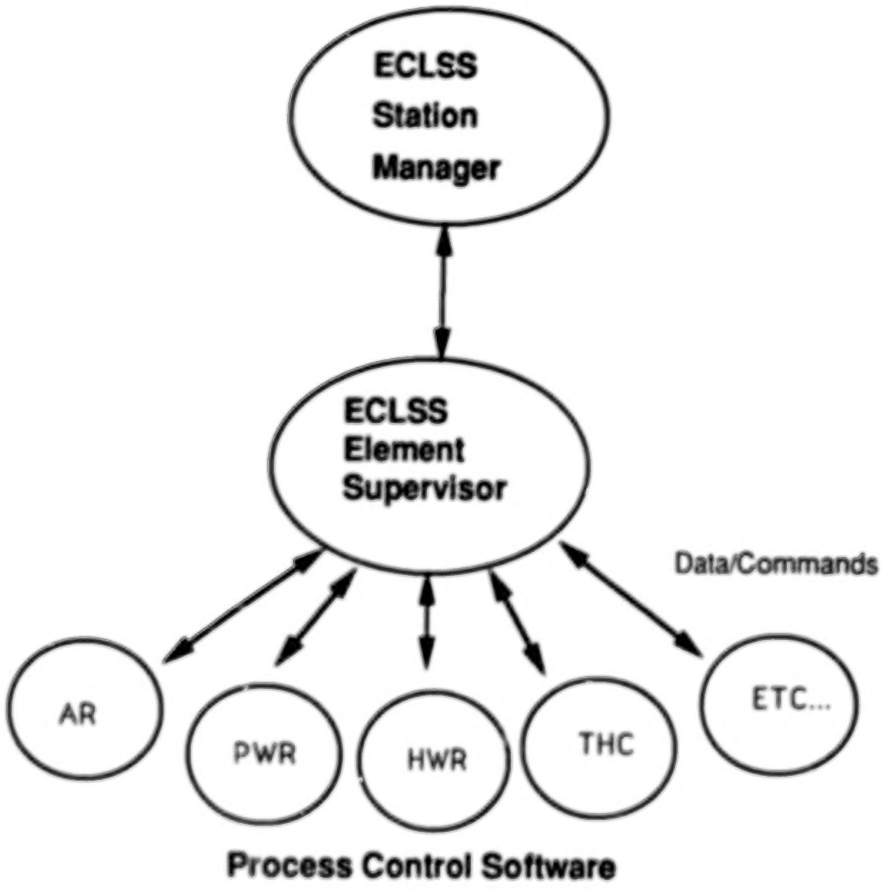


Figure 2 - ECLSS SOFTWARE FUNCTIONAL SCHEMATIC

Application Evaluation / Hooks and Scars Analysis

The preliminary research task, performed by the University of Alabama in Huntsville (UAH), has the main goal of laying the foundation for more extensive analysis and prototyping later, and to drive out software hooks[1] and hardware scars[2] early. It is expected to be an iterative task as the design and testing of ECLSS continues.

Application Evaluation / Hooks and Scars Analysis (figure 3) begins by analyzing the ECLSS domain . As the ECLSS is currenily in the preliminary design stage, our knowledge is generated from three general sources:

- Applicable Space Station Freedom documentation such as the ECLSS, DMS, OMS, Architecture Control Documents (ACD's), Contract End Item Specifications, ECLSS component test plans, etc.

- Conference reports on environmental control using knowledge based systems, plus papers describing other past work in automation of environmental control systems

- Interviews with ECLSS test and design engineers, scientists, and doctors

Our initial task was to gather and analyze promising documentation. The UAH application/hooks and scars analysis team, consisting of environmental, chemical, process control, and artificial intelligence engineers, then analyzed each document, determining areas in need of advanced automation and the resulting hooks and scars.

The ECLSS integration support contractor, McDonnell Douglas of Huntsville has been instrumental in supporting NASA ECLSS engineers to develop test plans and simulations for ECLSS components. As part of Phase 1 of this project, McDonnell Douglas was also employed to deliver ECLSS simulations and also write a brief report covering ECLSS instrumentation and possible knowledge based applications. This document was folded into the Domain analysis.

A "divergent thinking" approach was used in the initial listing of candidate applications (1). All the software functions found in ECLSS were listed in order to insure that all were considered. Boeing, the MSFC prime contractor for ECLSS, derived data flow diagrams in their software requirements generation. These were used to list and describe the functions of the baselined system. A preliminary application candidate list is shown in figure 4.

Also, new applications were added which were derived from interviews with ECLS system engineers and scientists. This list will serve as a broad groundwork for future iterations of application analysis.

[1] **software hooks** - an alteration to the Initial Operating Configuration software which will allow easier (less expensive) transition to more advanced software at a later date.
example: providing visibility of all ECLSS subsystem sensor data at the Global ECLSS Manager

[2] **hardware scars** - an alteration to the IOC hardware which will allow easier transition to more advanced hardware and automation capabilities at a later date.
example: providing a connection for a water quality monitor (to be added in the future) at the THC condensate output to isolate bugs in the water to a THC component.

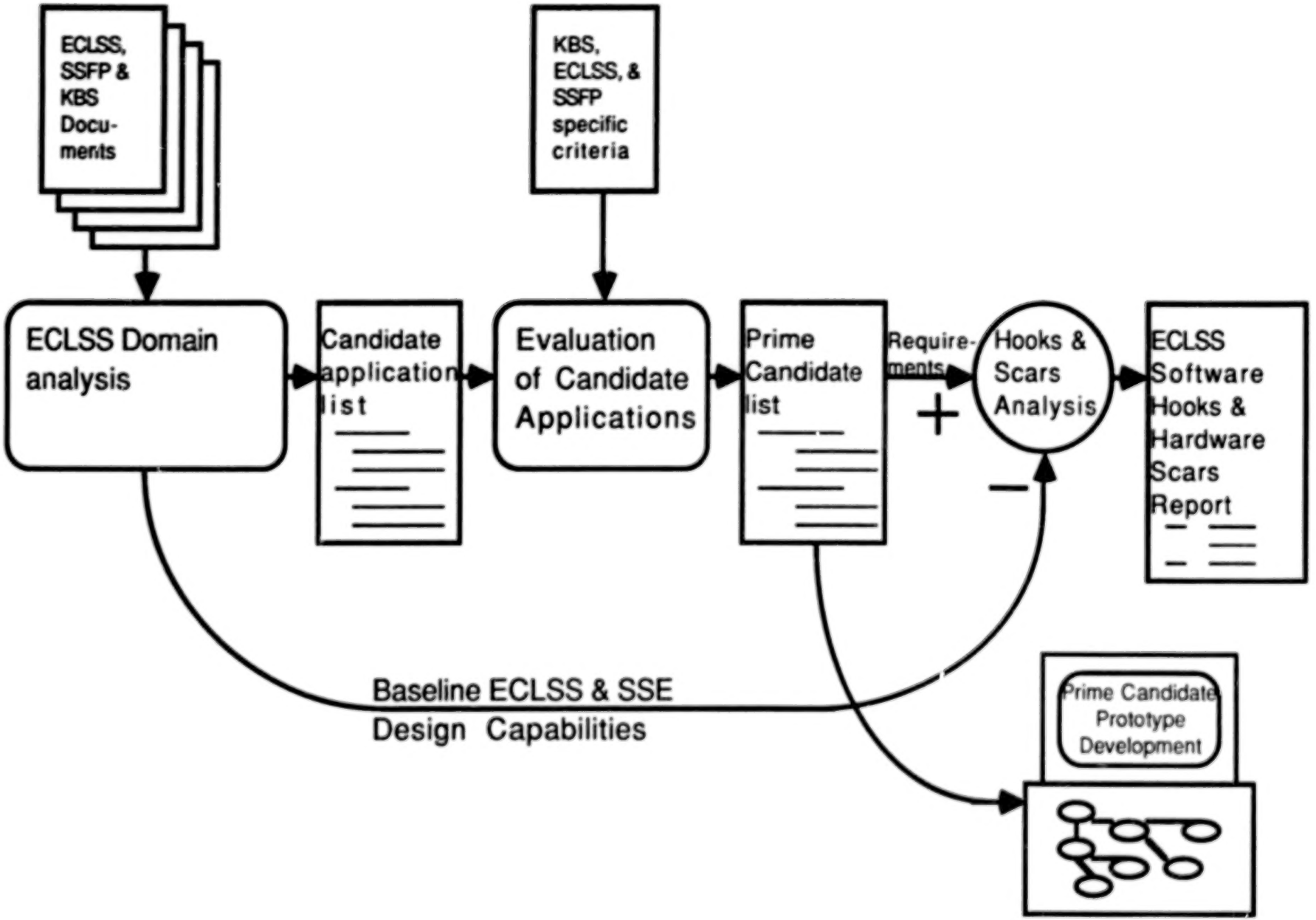


Figure 3 - Application Evaluation / Hooks and Scars Analysis Functional Schematic

I. ECLSS Station Manager

ECLSS system level control and FDIR software

. . .

II. ECLSS Element Manager

A. ECLSS Element Support Software

1. Fault Detection and Isolation
2. Performance and Trend Analysis
3. Verify Commands
4. Inhibit Commands
5. Activate Processes
6. Process Display Data

B. Temperature and Humidity Control (THC) Subsystem Software

. . .

C. Air Revitalization (AR) Subsystem Software

. . .

D. Atmosphere Control and Supply (ACS) Subsystem Software

. . .

E. Fire Detection and Suppression (FDS) Subsystem Software

. . .

F. Waste Management (WM) Subsystem Software

. . .

G. Water Recovery and Management (WRM) Subsystem Software

1. Process Potable Water
   a. Monitor Subsystem Status
   b. Monitor and Control Recieving Tank and Pump
   c. Monitor and Control the Purification Process
   d. Monitor and Control Potable Water Storage
   e. Monitor the Water Quality
2. Process Water Quality Sensor Data
3. Process Hygiene Water
4. Process Urine
5. Set WRM Limits

III. Real-time Process Control Software

A. Potable Water Processing Control

. . .

Figure 4 - Overview of Preliminary Candidate List

Evaluation criteria was derived for input into the prime candidate evaluation phase. Criteria for evaluation of knowledge based system (KBS) applications are well defined (1). An example of this KBS criteria is: "Is there an expert available? " These were used, but we have not limited ourselves to KBS implementations of advanced automation. We generated more evaluation criteria in the areas of: 1) ECLSS significance (ex: "has the need for advanced automation in this area been voiced by ECLSS engineers?"), and 2) Feasibility ("do we have enough time and money to complete a convincing demo of this application, or should we just analyze this application for later consideration?"). These criteria were used to select the prime candidate applications for which hooks and scars analysis will be performed. An outline of our prime candidate list, which we are in the process of developing, is given in figure 5.

Although the application evaluation plan (figure 3) calls for comple evaluation of all candidate applications before beginning a hooks and scars analysis, we found the best approach was to follow this procedure with one application at a time. For instance, some of our researchers at UAH are chemical and process control experts working on the Water Recovery Management's Potable Water Recovery (PWR) function. We found it best to explore the potable water loop in detail, finding prime candidate applications for advanced automation in this area and getting a good set of hooks and scars Little automation is currently planned in PWR. We will go back for a closer look into Hygiene Water Recovery and Air Revitalization later in Phase 1, after doing our procedural groundwork with PWR.

Having completed our domain analysis, we have done extensive analysis of application evaluation and hooks and scars for the Potable Water Recovery function. We plan to continue work on application analysis for the rest of the ECLSS regenerative functions, with more detail and hooks and scars requirements. Figure 6 is an illustration of the overall plan steps, pointing out our present position.

After phase 1 is complete (scheduled completion is the end of October), an engineering firm will develop prototype advanced automation software, picked from the prime candidate list and using the described requirements of phase 1. We will prove our automation concepts using system hardware and a software structure modeled on that of baselined ECLSS.

Application Examples

Two areas in which the ECLSS could be more autonomous are in process control and fault detection/prevention, and at the system level - supervising the overall operation of the regenerative loops. Advancement of the automation concepts in these two areas of ECLSS are illustrated. First, the potable water recovery system will be described along with a plan for increased automation. Second, the Meta-regenerative FDI function will be overviewed. These examples have not been picked for prototyping, but are prime candidates in our evaluation.

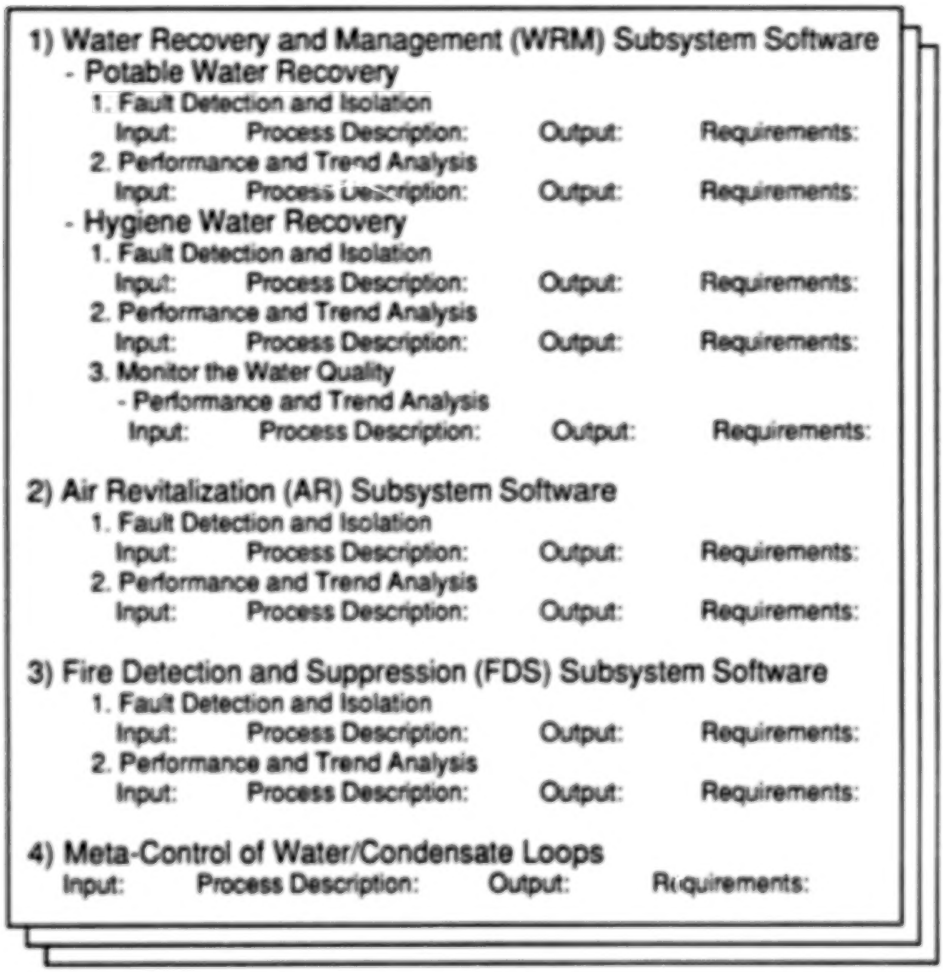


Figure 5 - Structure of Prime Candidate List

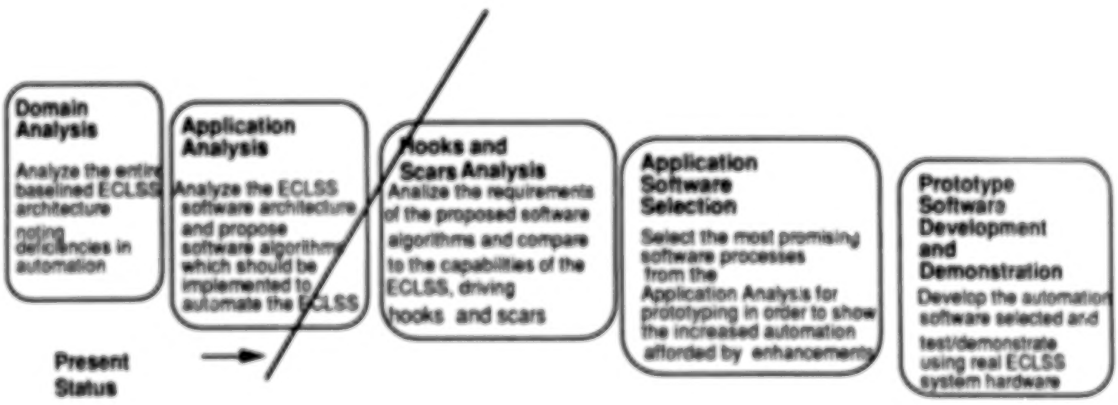


Figure 6 - Overview of Entire Task with Present Status

Potable Water Recovery Advanced Automation

The potable water recovery system, its control algorithm, and an overview of its advanced control algorithm is illustrated in figure 7. The system input is an inlet tank which gets its water from two sources, the THC condensate and the water by-product of the BOSCH carbon dioxide reduction subsystem.

This water is sent through a multifiltration device to an outlet tank which is checked using three monitoring methods for impurities:

1. The Process Control Water Quality Monitor checks in real time for levels of Iodine, Total Organics, PH, and Temperature.

2. A batch qualitive assessment is performed by the crew using a sample taken from the product water storage tank.

3. A batch qualitive assessment is performed on the ground using a sample brought back in the logistics module.

As the Potable Water Recovery components are being developed, so will the control scheme. Described in figure 7 is the baselined control scheme which uses the Water Quality Monitor data to determine in real time if the water is fit to drink. If its measurements meet specifications, the water is used, otherwise, it is sent back through multifiltration. The batch crew assessment is only to be used in off-nominal situations when there is not enough time to allow the ground assessment. The ground assessment will be used to match the readings of the WQM with the actual constituents of the water.

This control algorithm relies heavily on nominal processing of the water by the multifiltration device. This method may be sufficient for the initial operations of the Space Station Freedom, but advanced automation algorithms must be employed to reduce the crew and ground maintenance time. For instance, if an anomaly (a bug or foriegn chemical) gets in the water which multifiltration cannot remove, the water must be removed from the loop and it will be very difficult to determine its origin.

A general advanced control algorithm should not only use more knowledge in feedback decision but should attempt, if the water is contaminated, to determine the source of the contamination. Connection graphs, simulations, and trend data may be used to increase the intelligence of the control and fault detection algorithm. Also an expert system could be used in the ECLSS ground sustaining engineering facility to help correlate the sensor readings from the water quality monitor to the batch qualitive analysis of the water (2). This expert system would reduce analysis time on the ground and store valuable knowledge which will be acquired from a medical doctor or chemist working on ECLSS sustaining engineering.

The hygiene and air revitalization process control systems will be inspected for insufficient control algorithms and fault detection/isolation in a similar manner. We will drive higher fidelity control algorithms and hooks and scars for their later implementation.

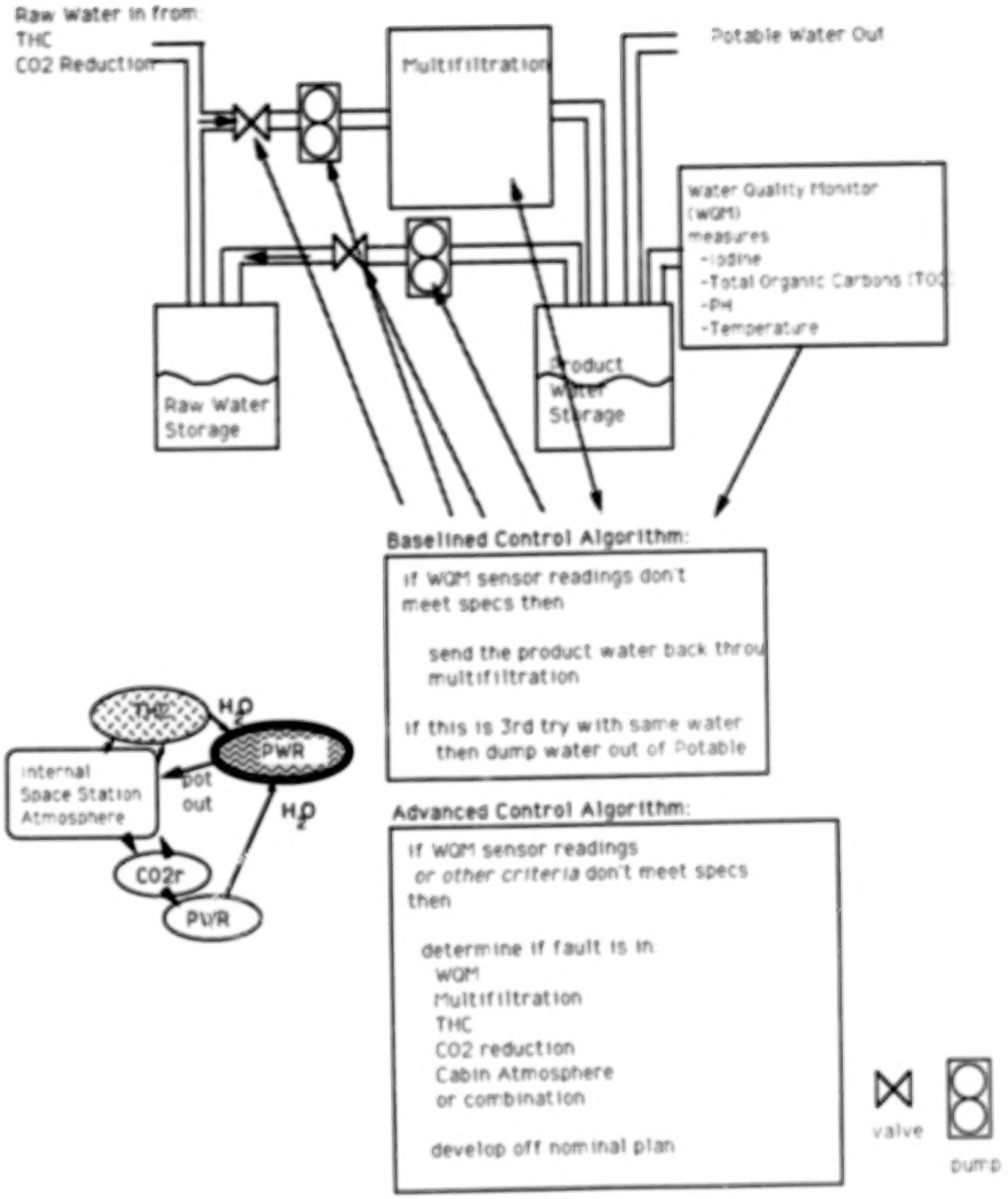


Figure 7 - Potable Water Recovery Advanced Automation

Meta-Regeneration Automation

The Meta-Regeneration Automation function is illustrated in figure 8. It will take as inputs the necessary data from the flight and ground analysis functions, and influence the control scheme in order to detect/predict/prevent a fault. A portion of the baseline software will most likely be used to serve this function, but as of yet it is not called out in the design. The ECLSS developers currently maintain that the subsystems are isolated enough that such an overall control intelligence is unnecessary, or at least minimal. In ECLSS extended testing, these system level requirements will become apparent - even though each subsystem is working nominally, instabilities may occur in a water or air regeneration loop. The initial operating configuration (IOC) of ECLSS may depend heavily on sustaining engineering analysis to detect these conditions and update the operating configuration. Hooks and scars in the IOC design should allow easy implementation of advanced automation software in this area.

This advanced software may take the form of a blackboard system or knowledge base system which has access to all the required data. These design decisions, and an operating prototype may be developed and tested in ECLSS subsystem testing. This system should be able to grow as more knowledge concerning the interaction of these processes is accumulated.

Conclusion

We developed a straightforward plan for advancing the automation of the Environmental Control and Life Support System. It was found that our group has expertise in a particular area of the ECLSS, the potable water recovery (PWR) subsystem, so we went into more detail in this area, using it as a case study for finding prime candidate applications and hooks and scars. We have yet to complete our application analysis for the entire ECLSS regeneration systems, and produce a hooks and scars analysis for these systems.

We have determined two examples of where the automation of the ECLSS could definitely be improved. One area is in the low level process control algorithms, the other is in the higher level of overall ECLS system automation. The baselined control algorithms of these systems will most assuredly change during system testing, but the groundwork of phase 1 will continue to be a valuable resource in prototype development and demonstration phases.

This preliminary analysis will prove beneficial in phase II and III, where actual prototypes will be developed by the engineering contractor. When these prototypes are being tested against ECLSS hardware and prototype software, the groundwork developed in phase I will minimize data searches and tracing of requirements.

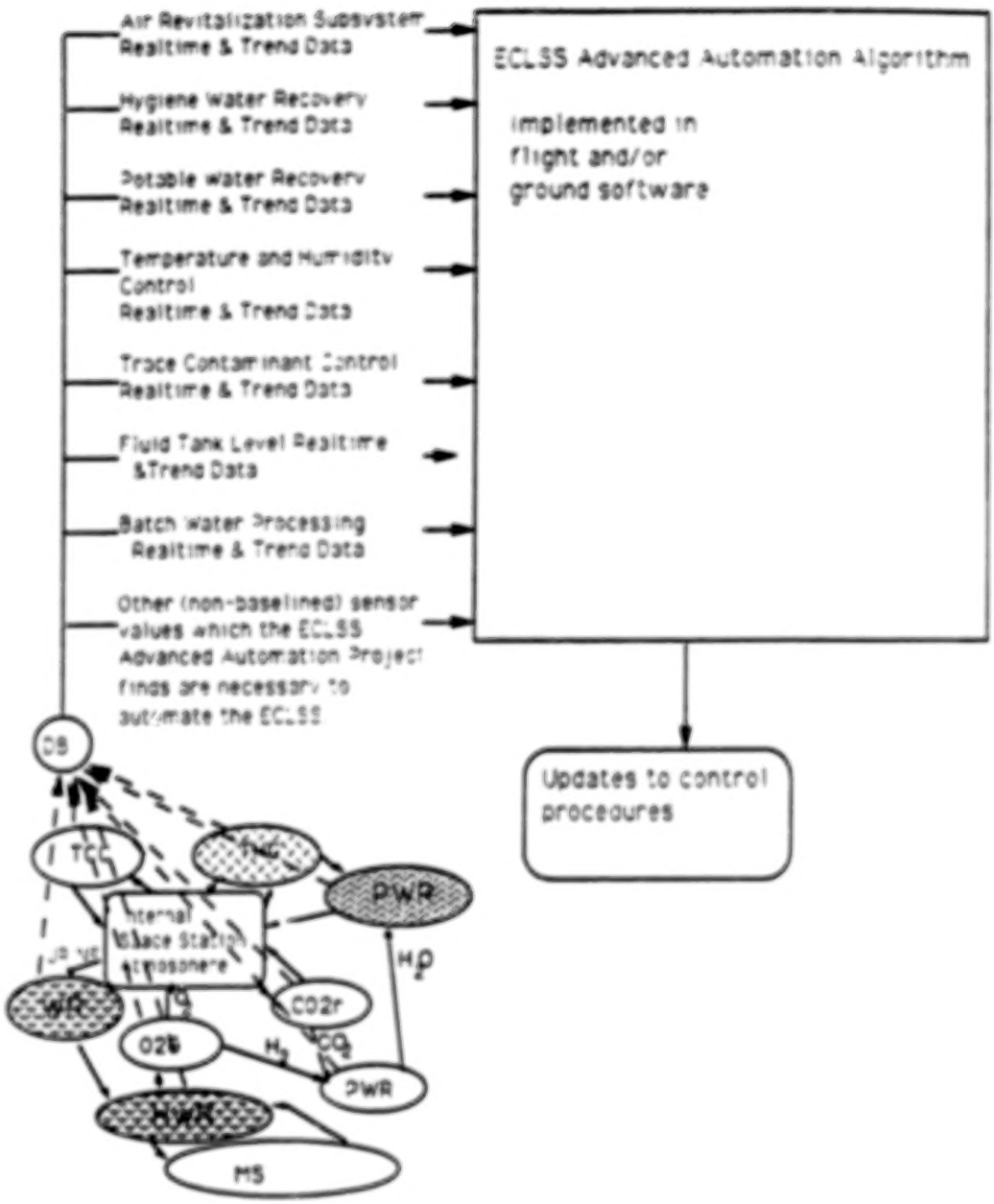


Figure 8 - Meta-Regeneration Automation

References

In addition to the reports listed below, some forty documents have been accumulated during the ECLSS domain analysis which were not explicitly referenced in this paper. A summarized list of these documents will be made available at the workshop.

1. Slagle, James R. and Wick, Michael R., "A Method For Evaluating Candidate Expert System Applications," AI Magazine, Palo Alto, CA, Winter 1988, pages 44 -53.

2. McDonnell Douglas Space Systems Company, Huntsville Division, "ECLSS Integration Analysis, Requirements Analysis of a Knowledge Base System (KBS) for the Space Station Environmental Control and Life Support System (ECLSS)," MDC W5184, MSFC Contract NAS8-36407, MSFC, AL, May 1989.

3. Dewberry, Brandon S., "Computer Control of the Space Station Environmental Control and Life Support System," MSFC Internal Memorandum for Record, MSFC, AL, April, 1988.

4.Architectural Control Document - Environmental Control and Life Support System, SSP 30262, February 15, 1989

Monitoring and Control of Spacecraft Systems Using Procedural Reasoning

Michael P. Georgeff[*]
Australian AI Institute
1 Grattan Street
Carlton, Victoria 3053
Australia

François Félix Ingrand[†]
Artificial Intelligence Center
SRI International
333 Ravenswood Avenue
Menlo Park, California 94025

Abstract

This paper describes research[‡] concerned with automating the monitoring and control of spacecraft systems. In particular, the paper examines the application of SRI's Procedural Reasoning System (PRS) to the handling of malfunctions in the Reaction Control System (RCS) of NASA's space shuttle. Unlike traditional monitoring and control systems, PRS is able to reason about and perform complex tasks in a very flexible and robust manner, somewhat in the manner of a human assistant. Using various RCS malfunctions as examples (including sensor faults, leaking components, multiple alarms, and regulator and jet failures), it is shown how PRS manages to combine both goal-directed reasoning and the ability to react rapidly to unanticipated changes in its environment. In conclusion, some important issues in the design of PRS are reviewed and future enhancements are indicated.

1 Introduction

As space missions increase in complexity and frequency, the automation of mission operations grows more and more critical. Such operations include subsystem monitoring, preventive maintenance, malfunction handling, fault isolation and diagnosis, communications management, maintenance of life support systems, power management, monitoring of experiments, satellite servicing, payload deployment, orbital-vehicle operations, orbital construction and assembly, and control of extraterrestrial rovers. Automation of these tasks can be expected to improve mission productivity and safety, increase versatility, lessen dependence on ground systems, and reduce demands for crew involvement in system control.

It is very important that any system designed to perform these tasks be as flexible, robust, and interactive as possible. At the minimum, it should be capable of responding to and diagnosing abnormalities in a variety of configurations and operational modes. It should be able to integrate information from various parts of the space vehicle systems and recognize potential problems prior to alarm limits being exceeded.

The system should suggest and execute strategies for containing damage and for making the system secure, without losing critical diagnostic information. It should be able to utilize standard malfunction handling procedures and take account of all the relevant factors that, in crisis situations, are easily overlooked. False alarms and invalid parameter readings should be detected, and alternative means for deducing parameter values should be utilized where possible.

In parallel with efforts to contain damage and temporarily reconfigure vehicle subsystems, the system should be able to begin diagnosis of the problem and incrementally adjust reconfiguration strategies as diagnostic information is obtained. The system should also be capable of communicating with other systems to seek information, advise of critical conditions, and avoid harmful interactions. Throughout this process, the system should be continually reevaluating the state of the space vehicle and should be capable of changing focus to attend to more serious problems should they occur.

Finally, the system should be able to explain the reasons for any proposed course of action in terms that are familiar to astronauts and mission controllers. It should be able to graphically display the system schematics, the procedures it is intending to execute, and the critical parameter values upon which its judgment is based.

Achieving this kind of behavior is well beyond the capabilities of conventional real-time systems. It requires, in contrast, mechanisms that can reason in a "rational" way about the state of the space vehicle and the actions that need be taken in any given situation. Moreover, the system should be both *goal directed* and *reactive*. That is, while seeking to attain specific goals, the system should also be able to react appropriately to new situations in real time. In particular, it should be able to completely alter focus and goal priorities as circumstances change. In addition, the system should be able to *reflect* on its own reasoning processes. It should be able to choose when to change goals, when to plan and when to act, and how to use effectively its deductive capabilities.

A number of system architectures for handling some of these aspects of real-time behavior have been recently proposed e.g., [Firby, 1987; Kaelbling, 1987; Hayes-Roth, 1985]. Some of these approaches are evaluated elsewhere [Georgeff and Ingrand, 1989; Georgeff and Lansky, 1987; Laffey *et al.*, 1988].

The system to be discussed in the paper is called a *Procedural Reasoning System* (PRS). It has been developed over a number of years at SRI International and has been reported, in part, in previous publications [Georgeff and Ingrand, 1989; Georgeff and Ingrand, 1988; Georgeff, 1988; Georgeff and Lansky, 1986a; Georgeff and Lansky, 1986b; Georgeff and Lansky, 1987].

2 Procedural Reasoning System

PRS is designed to be used as an embedded, real-time reasoning system. As shown in Figure 1, PRS consists of (1) a *database* containing current *beliefs* or facts about the world; (2) a set of current *goals* to be realized; (3) a set of *plans*, called knowledge areas (KAs), describing how certain sequences of actions and tests may be performed to achieve given goals or to react to particular situations; and (4) an *intention structure* containing all KAs that have been chosen for execution. An *interpreter* (or *inference mechanism*) manipulates these components, selecting appropriate plans based on the system's beliefs and goals, placing those selected on the intention structure, and executing them.

The system interacts with its environment, including other systems, through its database (which acquires new beliefs in response to changes in the environment) and through the actions that it performs as it carries out its intentions.

Goals and Beliefs

The beliefs of PRS provide information on the state of the space vehicle systems and are represented in a first-order logic. For

---

[*] Also a member of the Artificial Intelligence Center, SRI International, and the Center for the Study of Language and Information, Stanford University.

[†] Partially supported by the French research agency: Institut National de Recherche en Informatique et en Automatique (INRIA).

[‡] This research is supported by the National Aeronautics and Space Administration, Ames Research Center, under Contract No. NAS2-12121.

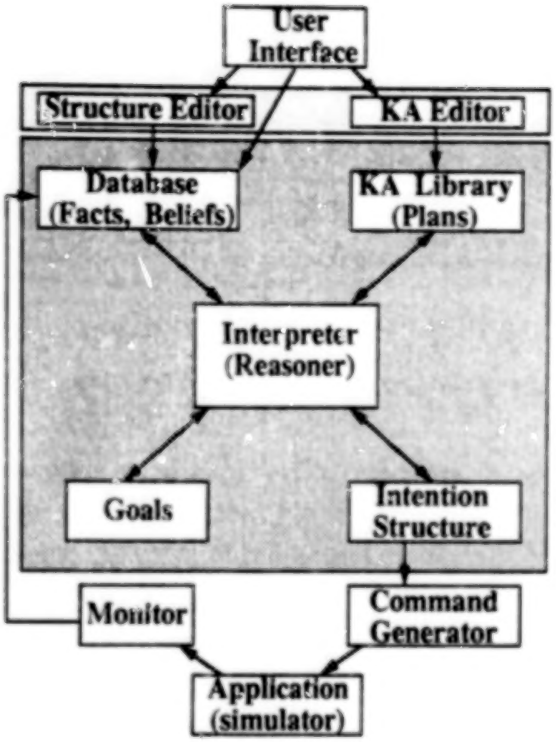


Figure 1: Structure of the Procedural Reasoning System

example, the fact that a particular valve, `v1` say, is closed could be represented by the statement `(position v1 cl)`.

The goals of PRS are descriptions of desired tasks or behaviors. In the logic used by PRS, the goal to achieve a certain condition C is written (! C); to test for the condition is written (? C); to wait until the condition is true is written (^ C); and to conclude that the condition is true is written (⇒ C). For example, the goal to close valve `v1` could be represented as `(! (position v1 cl)`, and to test for it being closed as `(? (position v1 cl))`.

Knowledge Areas

Knowledge about how to accomplish given goals or react to certain situations is represented in PRS by declarative procedure specifications called *Knowledge Areas* (KAs) (see, for example, Figure 10). Each KA consists of a *body*, which describes the steps of the procedure, and an *invocation condition*, which specifies under what situations the KA is useful and applicable. Together, the invocation condition and body of a KA express a declarative fact about the results and utility of performing certain sequences of actions under certain conditions [Georgeff and Lansky, 1986a].

The body of a KA can be viewed as a plan or plan schema. It is represented as a graph with one distinguished start node and possibly multiple end nodes. The arcs in the graph are labeled with the subgoals to be achieved in carrying out the plan. Successful execution of a KA consists of achieving each of the subgoals labeling a path from the start node to an end node. This formalism provides a natural and efficient representation of plans involving any of the usual control constructs, including conditional selection, iteration, and recursion.

The invocation condition contains a *triggering* part describing the *events* that must occur for the KA to be executed. Usually, these events consist of the acquisition of some new goals (in which case, the KA is invoked in a goal-directed fashion) or some change in system beliefs (resulting in data-directed or reactive invocation) and may involve both.

The set of KAs in a PRS application system not only consists of procedural knowledge about a specific domain, but also includes *metalevel* KAs; that is, information about the manipulation of the beliefs, goals, and intentions of PRS itself. For example, typical metalevel KAs encode various methods for choosing among multiple applicable KAs, modifying and manipulating intentions, and computing the amount of reasoning that can be undertaken, given the real-time constraints of the problem domain.

The Intention Structure

The intention structure contains all those tasks that the system has chosen for execution, either immediately or at some later time. These adopted tasks are called *intentions*. A single intention consists of some initial KA together with all the sub-KAs that are being used in attempting to successfully execute that KA. It is directly analogous to a *process* in a conventional programming system.

At any given moment, the intention structure may contain a number of such intentions, some of which may be suspended or deferred, some of which may be waiting for certain conditions to hold prior to activation, and some of which may be metalevel intentions for deciding among various alternative courses of action.

For example, in handling a malfunction in a propulsion system, PRS might have, at some instant, three tasks (intentions) in the intention structure: one suspended while waiting for, say, the fuel-tank pressure to decrease below some designated threshold; another suspended after having just posted some goal that is to be accomplished (such as interconnecting one shuttle subsystem with another); and the third, a metalevel procedure, being executed to decide which way to accomplish that goal.

Execution

Unless some new belief or goal activates some new KA, PRS will try to fulfill any intentions it has previously decided upon. This results in focussed, goal-directed reasoning in which KAs are expanded in a manner analogous to the execution of subroutines in procedural programming systems. But if some important new fact or goal does become known, PRS will reassess its current intentions and perhaps choose to work on something else. Thus, not all options that are considered by PRS arise as a result of means-end reasoning. Changes in the environment may lead to changes in the system goals or beliefs, which in turn may result in the consideration of new plans that are not means to any already intended end. PRS is therefore able to change its focus completely and pursue new goals when the situation warrants it. In many space operations, this may happen quite frequently as emergencies of various degrees of severity occur in the process of handling other, less critical tasks.

Multiple Systems

In some applications, it is necessary to monitor and process many sources of information at the same time. Because of this, PRS was designed to allow several instantiations of the basic system to run in parallel. Each PRS instantiation has its own data base, goals, and KAs, and operates asynchronously relative to other PRS instantiations, communicating with them by sending messages.

The system described above has been implemented on Symbolics 3600 Series LISP, Sun Series 3, and Mac Ivory machines. A more complete description of PRS can be found elsewhere [Georgeff and Ingrand, 1989; Georgeff and Ingrand, 1988].

3 The RCS Application

The system chosen for experimentation with PRS is the Reaction Control System (RCS) of the space shuttle. The system structure is depicted in the schematic of Figure 2 (left part). One of the aims of our research is to automate the malfunction procedures for this subsystem. A sample malfunction procedure is presented in Figure 3.

The RCS provides propulsive forces from a collection of jet thrusters to control the attitude of the space shuttle. There are three RCS modules, two aft and one forward. Each module contains a collection of primary and vernier jets, a fuel tank, an oxidizer tank, and two helium tanks, along with associated feedlines, manifolds, and other supporting equipment. Propellant flow, both

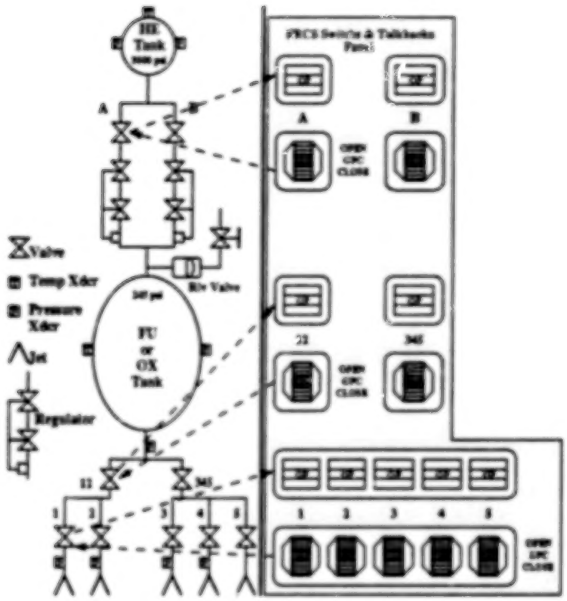


Figure 2: System Schematic for the RCS

fuel and oxidizer, is normally maintained by pressurizing the propellant tanks with helium.

The helium supply is fed to its associated propellant tank through two redundant lines, designated A and B. The pressure in the helium tanks is normally about 3000 psi; this is reduced to about 245 psi by regulators that are situated between each helium tank and its corresponding propellant tank. A number of pressure and temperature transducers are attached at various parts of the system to allow monitoring.

Each RCS module receives all commands (both manual and automatic) via the space shuttle flight computer software. This software resides on five general purpose computers (GPCs). Up to four of these computers contain redundant sets of the Primary Avionics Software System (PASS) and the fifth contains the software for the Backup Flight System (BFS). All of the GPCs can provide information to the crew by means of CRT displays.

The various valves in an RCS module are controlled from a panel of switches and talkbacks (Figure 2, right part). Each switch moves associated valves in *both* the fuel subsystem and the oxidizer subsystem.[2] Switches can be set to OPEN, CLOSE, or GPC, the last providing the GPCs with control of the valve position. The talkback provides feedback on the associated valve position. The talkback reading normally corresponds with the associated switch position, except when the switch is in GPC; in this case, the talkback shows whichever position the GPC puts the valve in. The talkbacks may not correspond if a valve has jammed or if the control or feedback circuit is faulty. If the valves in both the fuel and oxidizer subsystems do not move in unison, because of some fault, the talkback displays a barberpole.

As with most dynamic systems, transient faults are common. For example, in the process of changing switch position, there will be a short time (about 2 seconds) when the positions of the talkback and the switch will differ from one another. This is because it takes this amount of time for the actual valve to change its position. Furthermore, during this transition, the talkback will also pass through the barberpole position. Thus, a mismatched talkback and switch position or a barberpole reading does not always indicate a system fault.

[2]Because the two propellant subsystems are identical, only one system is represented in the left part of the figure.

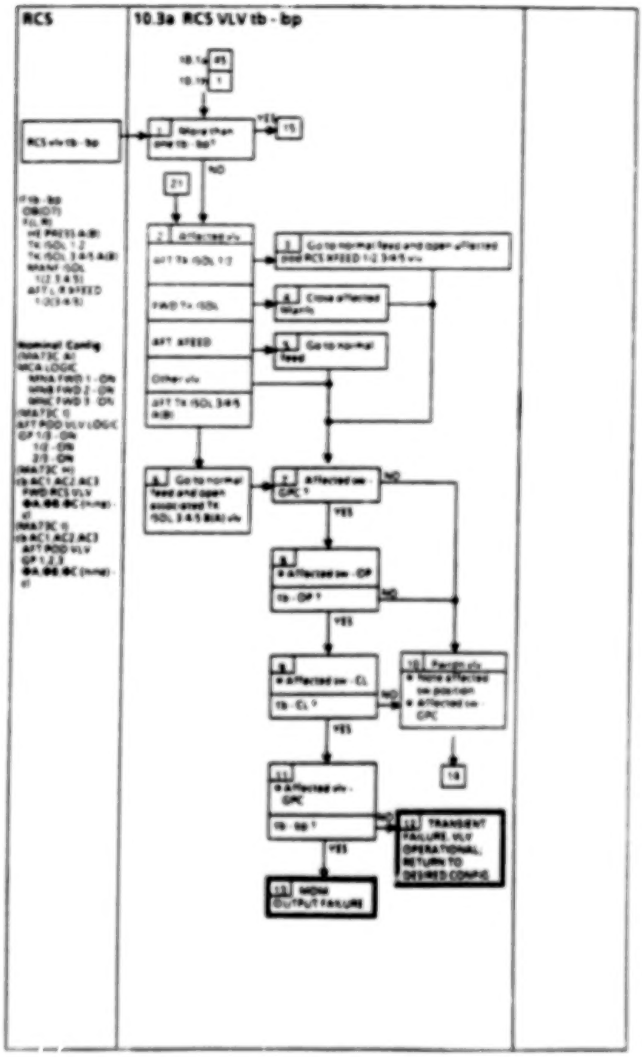


Figure 3: A RCS Malfunction Procedure

4 System Configuration

Two instances of PRS were set up to handle the RCS application. One, called **INTERFACE**, handles most of the low level transducer readings, effector control and feedback, and checks for faulty transducers and effectors. The other, called somewhat misleadingly **RCS**, contains most of the high-level malfunction procedures, much as they appear in the malfunction handling manuals for the shuttle. To test the system, a simulator for the actual RCS was constructed.

The complete system configuration is shown in Figure 4. Each of these parts is described in the following sections.

4.1 The Simulator

During operation, the simulator sends transducer readings and feedback from various effectors (primarily valves) to **INTERFACE** and communicates alarm messages as they appear on the shuttle system displays to **RCS**. The simulator, in turn, responds appropriately to changes in valve switch positions as requested by **INTERFACE**. The simulator can be set to model a variety of fault conditions, including misread transducers, stuck valves, system leaks, and regulator failures.

A future implementation of the system will be connected to the more sophisticated shuttle simulator used at Johnson Space Center.

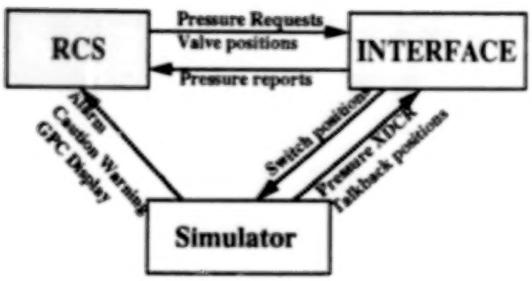


Figure 4: System Configuration

4.2 The RCS

The top-level PRS instantiation, RCS, contains most of the malfunction handling procedures as they appear in the operational manuals for the space shuttle. RCS takes an abstract view of the domain: it deals in pressures and valve positions, and does not know about transducers, switches, or talkbacks. For example, whenever RCS needs to know the pressure in a particular part of the system, it requests this information from INTERFACE, which is expected to deduce the pressure from its knowledge of transducer readings and transducer status. Similarly, RCS will simply request that INTERFACE moves a valve to a certain position, and is not concerned how this is achieved. In this way, RCS can represent the malfunction handling procedures in a clean and easily understandable way, without encumbering the procedures with various cross-checks and other details.

4.3 The INTERFACE

The PRS instantiation INTERFACE handles all information concerning transducer readings, valve switches, and valve talkbacks. It handles requests from RCS for information on the pressures in various parts of the system and for rates of change of these values. Determination of this information can require examination of a variety of transducers, as readings depend on the status of individual transducers, their location relative to the region whose pressure is to be measured, and the connectivity of the system via open valves.

INTERFACE also handles requests from RCS to change the position of the valves in the RCS. This involves asking the astronaut to change switch positions, and waiting for confirmation from the talkback.

While doing these tasks, INTERFACE is continually checking for failures in any of the transducers or valve assemblies. When it notices such failures, it will notify the astronaut or mission controller and appropriately modify its procedures for determining pressures or closing valves. It will also consider the consequences of any failures, such as are prescribed in various flight rules for the shuttle.

5 Sample Interactions

In this section, we examine different scenarios illustrating the capabilities of PRS.

5.1 Changing Valve Position

The following example illustrates the capacity of the system to reason about more than one task at a time. Consider the situation where INTERFACE gets a request from RCS to close some valve, say `frcs-ox-tk-isol-12-valve` (Forward RCS, OXidizer TanK, one-two ISOLation VALVE). RCS achieves this by sending INTERFACE the message `(request RCS (!(position frcs-ox-tk-isol-12-valve cl)))`. Responding to this request, INTERFACE calls a KA that, in turn, asks the astronaut to place the switch corresponding to this valve in the closed position (see Figure 5). Once the astronaut has done this, INTERFACE will wait until the talkback shows the requested position and will then advise RCS that the valve has indeed been closed (Figure 5).

However, while this is taking place, INTERFACE will also notice that, just after the switch is moved to the closed position, there is a mismatch with the talkback indicator (which will still be showing open, because of the normal delay in the valve starting to move). Furthermore, a fraction of a second later, the talkback will move into the barberpole position, another indication that things could be wrong with the valve.

Each of these events will trigger a KA and thus initiate execution of a task (intention) that seeks to confirm that the talkback moves to its correct position within a reasonable time; Figure 6 shows the KA which monitors the barberpole position. At this point, the system is dealing with three different tasks, one responsible for answering the request, one checking the miscomparison between the switch and the talkback, and one checking for the barberpole position. Each of these last two tasks immediately suspend themselves (using the "wait-until" (`) operator) while awaiting the specified condition to become true.

For example, the task concerned with monitoring a talkback barberpole reading will suspend itself until either the positions of both the switch and the talkback agree, or 10 seconds elapses. When either of these conditions become true, the task (intention) will awaken and proceed with the next step. If the talkback is still in the barberpole position, the astronaut or mission controller will be notified of the problem. Otherwise, the KA *fails*, and simply disappears from the intention structure.

Notice that the KAs that respond to the request from RCS to change the valve position, that monitor for possible switch dilemmas, and that check the barberpole reading are all established as different intentions at some stage during this process. Various metalevel KAs must therefore be called, not only to establish these intentions, but to decide which of the active ones to work on next.

A typical state of the intention structure is shown in Figure 7. It shows a number of intentions in the system INTERFACE, ordered for execution as indicated by the arrows. The intention labeled `Meta Selector` is a metalevel KA (Figure 8). The other intentions include two that are checking potential switch problems (`Switch Dilemma (Barberpole)` and `Switch Dilemma (Closed)`) and one that is responding to the request to close the valve (`Open or Close Valve`). The metalevel intention, in this case, is the one currently executing. Although not clear from the figure, it has just created and ordered the new intentions resulting from the talkback and the barberpole problems.

5.2 Handling Faulty Transducers

In this scenario, we show how two PRS agents cooperate and control the execution of their intentions so as to handle faulty transducers and the resulting false warning alarms.

We will assume that transducer `frcs-ox-tk-out-p-xdcr` fails and remains jammed at a reading of 170 psi. This causes a number of things to happen. First, it causes a low-pressure alarm to be activated. This will will be noticed by the PRS instantiation RCS, which will immediately respond to the alarm by initiating execution of the KA (`Pressurization Alarm (Propellant Tank)`). This KA will, in turn, request a pressure reading from INTERFACE to ensure that the alarm is valid.

While this is happening, INTERFACE by itself has noticed that the two transducers on the oxidizer tank disagree with one another (in this case, the other transducer is reading the nominal value of 245 psi). This invokes a KA that attempts to determine which of the two transducers is faulty. It does this by first waiting a few seconds to ensure that the mismatch is not simply a transient, and then testing to see if one of the readings is outside normal limits. If so, it assumes this is the faulty transducer; this is indeed the procedure used by astronauts and mission controllers. Other KAs, capable of more sophisticated acts such as checking the values of downstream or upstream transducers, are used if there is no corresponding transducer with which to do the cross-check.

Notice what could happen here if one is not careful. Having more than one thing to do, INTERFACE could decide to service the request for a pressure reading for the suspect tank. If it does so, it will simply average the values of the two transducer readings (yielding 207 psi) and advise RCS accordingly. Clearly, this is not

Open or Close Valve

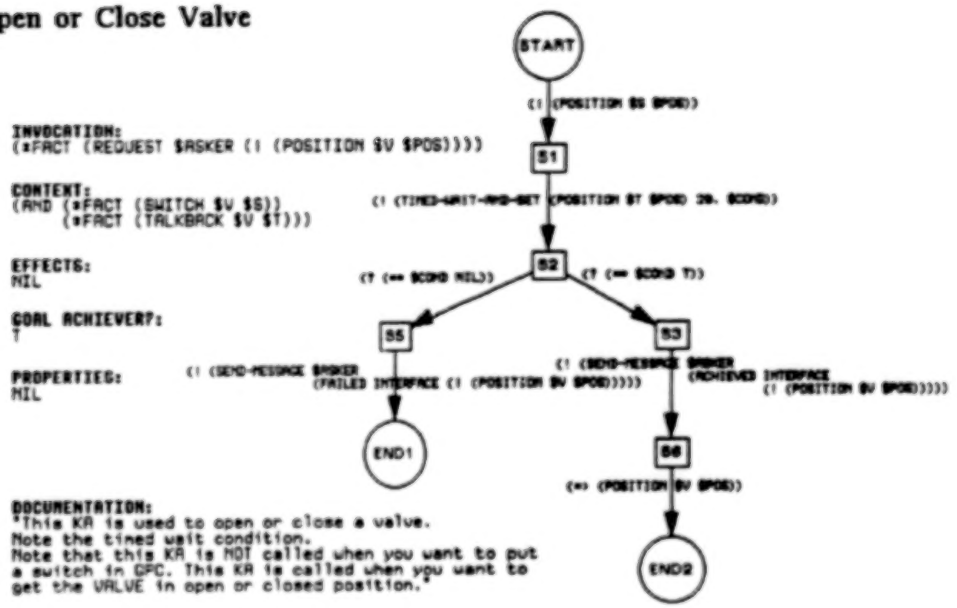


Figure 5: KA for Closing a Valve

Switch Dilemma (Barberpole)

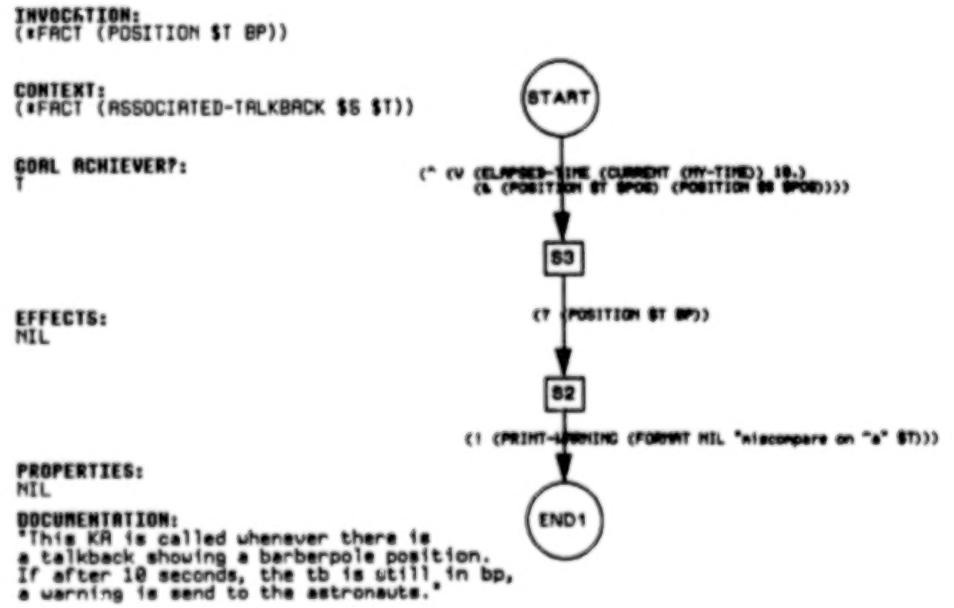


Figure 6: KA for Monitoring Talkback in Barberpole

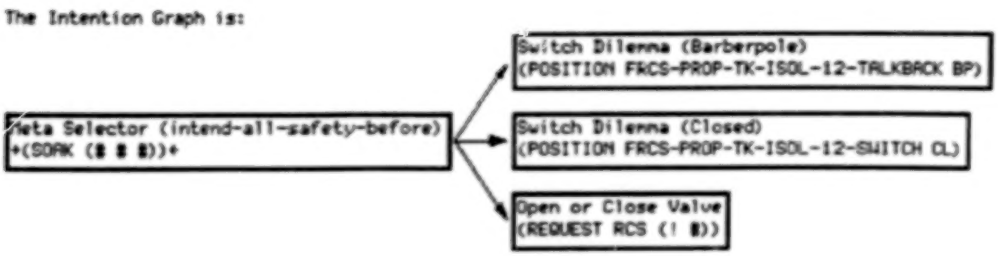


Figure 7: Intention Structure during Switch Operation

Meta Selector (intend-all-safety-before)



Figure 8: The Metalevel KA **Meta Selector**

what we want to happen: any suspect parameter readings should be attended to before servicing requests that depend on them.

In the examples we have considered, it has been sufficient to handle such problems with a relatively simple priority scheme. We first ascribe the property of being a so-called *"safety handler"* to all those KAs that should be executed at the earliest possible time. Then we design the metalevel KA that chooses between potentially applicable KAs to order all safety handlers for execution prior to other intentions. In the example given above, the KA that detects the faulty transducer is a safety handler, and thus is executed prior to servicing the request from **RCS**. When **INTERFACE** eventually gets around to servicing the request from **RCS**, it disregards the faulty transducer reading and thus advises **RCS** that the pressure is 245 psi. **RCS** then determines that the alarm was activated in error and that the pressure is within normal operating range.

Even with all this going on, other things are happening within the **INTERFACE** system. For example, the fact that the transducer is determined to be bad, together with the fact that it is the very transducer that informs the shuttle computers of overpressurization problems, causes the invocation of another KA. This KA reflects a flight rule that states that overpressurization protection is lost while the transducer is inoperative.

As before, metalevel KAs are invoked to determine which KAs to adopt as intentions and how to order them on the intention structure. The development of the intention structure during this process is shown in Figure 9.

5.3 Failed Regulator

Let's now consider the operation of the top-level PRS instantiation, **RCS**. The case we first examine occurs when the regulator on the feed line between the helium tank and its associated propellant tank fails. In this example, we will assume that the **frcs-fu-he-tk-A-reg** has failed. We will focus primarily on **RCS** (**INTERFACE** is, of course, working away during this process as discussed above).

The first thing that happens when the regulator fails is that pressures throughout the fuel subsystem begin to rise. When they exceed the upper limit of 300 psi, certain caution-warning (cw) alarms are activated. These events trigger the execution of a KA that attempts to confirm that the system is indeed overpressurized.

Note that this process is more complicated than it first appears. The high transducer readings that gave rise to the caution-warning alarm will also trigger KAs in the PRS system **INTERFACE**. These KAs will proceed to verify that the corresponding transducers are not faulty (as described in subsection 5.2); that is, that the reading of the transducers is indeed accurate. While doing this, or after doing this, **INTERFACE** will get a request from **RCS** to advise the latest pressure readings. If **INTERFACE** is in the process

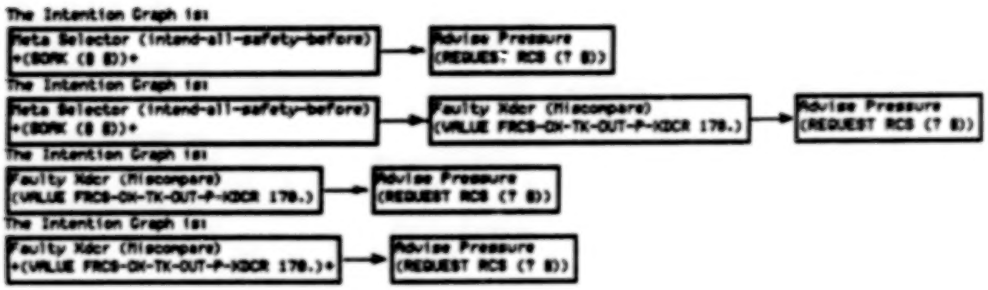


Figure 9: Intention Structure Development

of checking the transducers, it will defer answering this request until it has completed its evaluation of transducer status. But eventually it will return to answering the request and, in the case we are considering, advise that the pressure is indeed above 300 psi.

On concluding that the system is overpressurized, another KA (**Overpressurized Propellant Tank**) is activated and this, eventually, concludes that the A regulator has failed (see Figure 10). Note that this KA establishes subgoals to close both the A valve and the B valve, as there are cases when both are open. For the A valve, this involves a request to **INTERFACE** as discussed above. However, for the B valve, the system notices that the B valve is already closed. Thus, its goal is directly achieved without the necessity to perform any action or request.

The final goal of this KA activates another KA that opens the valve of the alternate regulator (B). Having opened the valve, it is desirable to then place it under the control of the on-board computers. However, this cannot be done until the pressure in the system drops below 300 psi, as otherwise the GPC will automatically shut the valve again. Thus, the malfunction handling procedures specify that the astronaut should wait until this condition is achieved before proceeding to place the valve switch in the GPC position. **RCS** achieves this by asking **INTERFACE** to monitor the pressure and advise it when it drops below 300 psi. While waiting for an answer, the task is suspended, and **RCS** gets on with whatever else it considers important.

When the pressure eventually drops below that threshold, the task (intention) is awakened, and execution continued. Thus, the valve switch is finally placed in the GPC position and the overpressurization problem resolved.

5.4 Isolating a System Leak

Let's assume that there is a leak in the RCS. Usually, the leak will cause a pressure drop in the system that will trigger a caution-warning alarm. The KA that responds to this alarm will first try to differentiate between a failed regulator and a leak in the system. If it determines that the system has a leak, it will then establish the goal to isolate that leak. This, in turn, triggers another KA that first attempts to secure the system This involves requesting that the astronauts close all valves in the leaking system.

Again, the PRS system **INTERFACE** will, throughout each process of closing a valve, check that the valve has indeed closed and that the corresponding talkbacks are registering closed.

As soon as the system has been secured, PRS identifies the leaking section by checking for decreasing pressure in each section of the RCS in turn.

6 Conclusion

The experiments described above provided a severe and positive test of the system's ability to operate proficiently in real time, to weigh alternative courses of action, to coordinate its activities, and to modify its intentions in response to a continuously changing environment. In addition, PRS met every criterion outlined by Laffey et al. [1988] for evaluating real-time reasoning systems: high performance, guaranteed response, temporal reasoning capabilities, support for asynchronous inputs, interrupt handling, continuous operation, handling of noisy (possibly inaccurate) data, and shift of focus of attention.

We believe that the following features of PRS played an important role in achieving these results.

Procedural reasoning: The representation of procedural knowledge using KAs is a very powerful way to describe the actions and procedures that should be executed to accomplish specific goals or to respond to certain critical events. One essential feature of the representation is that the elements of these procedures are described in terms of their *behaviors* rather than in terms of arbitrarily named actions or subroutines. For example, to achieve the goal "close all affected manifolds," it is essential to be able to reason about the intended set of manifolds and how the goal is then to be achieved; a call to a specialized procedure for every variant of this goal is simply too complex and too prone to error. Furthermore, a descriptive (declarative) representation of goals provides robustness as different procedures (KAs) can be used to accomplish the goal depending on the mode of operation, the availability of resources, or the time required to perform the task. Moreover, because the purpose of each step in the procedure is so represented, other processes can independently decide how to achieve their own goals without thwarting that plan; indeed, they may even decide to assist.

Reactive and goal-directed reasoning: The capability of being simultaneously data- and goal-driven is a critical feature of PRS. PRS provides goal-driven reasoning when explicit goals must be achieved, such as closing a valve, or repressurizing a system. At the same time, the reactive capabilities of PRS allow it to respond to critical events that occur, even when PRS is itself attending to some other task. This capability of reacting to new events makes the system highly adaptive to situation changes: any plan can be interrupted and reconsidered in the light of new incoming information.

Real-time reasoning: One of the most important measures in real-time applications is reaction time; if events are not handled in a timely fashion, the process can go out of control. PRS has been designed so that such a guarantee can be furnished. Although PRS can execute complex conditional plans, the inference mechanism used in PRS guarantees that any new event is noticed in a bounded time [Georgeff and Ingrand, 1989; Georgeff and Ingrand, 1988]. While the system is executing any procedure, it monitors new incoming events and goals. Given that the real time behavior of the metalevel KAs used in a PRS application can be analyzed, the user can prove that his application can operate in real time: any new event is taken care of in a bounded time.

Reasoning about multiple tasks: The intention structure used in PRS enables the system to attend to more than one task at a time. These multiple intentions are usually tightly coupled and the order in which they are executed can be very important. Some may require immediate execution on the basis of urgency; others may have to be scheduled later than others because they depend on the results produced by the earlier tasks. Potential interactions among concurrently executing intentions can also be critical in deciding the most appropriate ordering of tasks. PRS provides the mechanisms to examine and manipulate the intention structure directly; the user can thus specify any kind of priority or scheduling scheme desired.

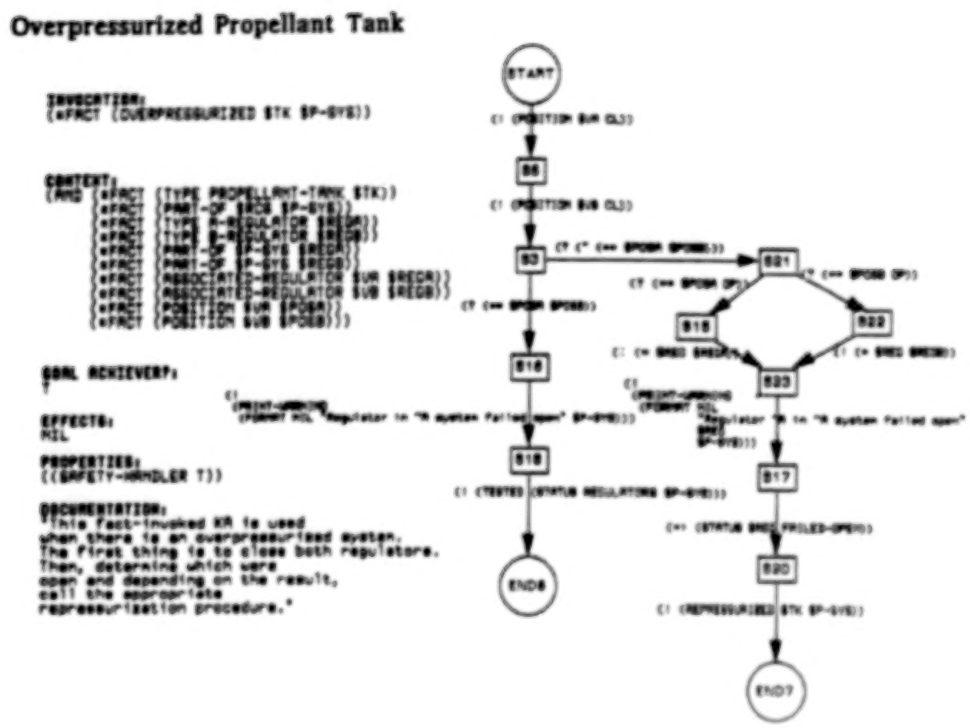


Figure 10: KA for Overpressurized Propellant Tank

Metalevel reasoning: The provision of metalevel KAs allows the system to control its problem solving strategies in arbitrarily sophisticated ways. These metalevel KAs follow the same syntax and semantics as application KAs, except that they deal with the control of the execution of PRS itself. Thus one can write metalevel KAs that can reason efficiently and effectively about the problem solving process being used. For example, one can have a KA to control in which order the applicable KAs are going to be executed. In the example presented in the subsection 5.2, the metalevel KA makes sure that the system carries on the testing task before the pressure update task, thus allowing the false alarm to be correctly recognised. Similarly, one can use metalevel KAs to choose among different ways to perform a given task, or how best to meet the real-time constraints of the domain given information on the expected time required for task execution.

Distributed reasoning: PRS is designed for distributed operations. Thus, different instances of PRS can be used in any application that requires the cooperation of more than one agent. The different PRS agents run asynchronously; their activity is therefore unconstrained a priori by that of their colleagues. A message passing mechanism is provided to make possible communication between the different PRS agents as well as with external modules such as simulators or monitors.

A number of critical research problems remain to be solved before the system will be reliable enough for use in actual space operations. The system is currently being extended to cover all malfunction handling procedures and flight rules concerning the RCS and is to be tested against the main shuttle simulators at Johnson Space Center in future work.

Acknowledgments

Pierre Bessiere, Amy Lansky, Anand Rao, Lorna Shinkle, Joshua Singer, Mabry Tyson, and Dave Wilkins helped in the development of PRS and extended the implementation as needed. Matthew Barry and Janet Lee helped in the development of the RCS application and Oscar Firschein provided constructive criticism of the paper.

References

[Firby, 1987] R. J. Firby. An investigation into reactive planning in complex domains. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 202–206, Seattle, Washington, 1987.

[Georgeff and Ingrand, 1988] M. P. Georgeff and F. F. Ingrand. Research on procedural reasoning systems. Final Report, Phase 1, for NASA Ames Research Center, Moffet Field, California, Artificial Intelligence Center, SRI International, Menlo Park, California, October 1988.

[Georgeff and Ingrand, 1989] M. P. Georgeff and F. F. Ingrand. Decision-making in an embedded reasoning system. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, Michigan, 1989. To appear.

[Georgeff and Lansky, 1986a] M. P. Georgeff and A. L. Lansky. Procedural knowledge. *Proceedings of the IEEE Special Issue on Knowledge Representation*, 74:1383–1398, 1986.

[Georgeff and Lansky, 1986b] M. P. Georgeff and A. L. Lansky. A system for reasoning in dynamic domains: Fault diagnosis

on the space shuttle. Technical Note 375, Artificial Intelligence Center, SRI International, Menlo Park, California, 1986.

[Georgeff and Lansky, 1987] M. P. Georgeff and A. L. Lansky. Reactive reasoning and planning: An experiment with a mobile robot. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, Seattle, Washington, 1987.

[Georgeff, 1988] M. P. Georgeff. An embedded real-time reasoning system. In *Proceedings of the 12th IMACS World Congress*, Paris, France, 1988.

[Hayes-Roth, 1985] B. Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence*, 26(3):251–321, 1985.

[Kaelbling, 1987] L. P. Kaelbling. An architecture for intelligent reactive systems. In *Reasoning about Actions and Plans: Proceedings of the 1986 Workshop*, pages 395–410. Morgan Kaufmann, Los Altos, California, 1987.

[Laffey *et al.*, 1988] T. J. Laffey, P. A. Cox, J. L. Schmidt, S. M. Kao, and J. Y. Read. Real-time knowledge-based systems. *AI Magazine*, 9(1):27–45, 1988.

Knowledge Representation to Support Reasoning Based on Multiple Models

April Gillam, Jorge P. Seidel
Computer Science Laboratory
The Aerospace Corporation
Los Angeles, CA 90009
and Alice C. Parker
Dept. of Electrical Engineering - Systems
University of Southern California

This work was supported by the
Aerospace Sponsored Research Program.

1 ABSTRACT

Model Based Reasoning is a powerful tool used to design and analyze systems, which are often composed of numerous interactive, interrelated subsystems. Models of the subsystems are written independently and may be used together while they are still under development. Thus the models are not static. They evolve as information becomes obsolete, as improved artifact descriptions are developed, and as system capabilities change. We are using three methods to support knowledge/data base growth, to track the model evolution, and to handle knowledge from diverse domains. First, the representation methodology is based on having pools, or types, of knowledge from which each model is constructed. In addition information is explicit. This includes the interactions between components, the description of the artifact structure, and the constraints and limitations of the models. The third principle we have followed is the separation of the data and knowledge from the inferencing and equation solving mechanisms. This methodology is used in two distinct knowledge-based systems: one for the design of space systems and another for the synthesis of VLSI circuits. It has facilitated the growth and evolution of our models, made accountability of results explicit, and provided credibility for the user community. These capabilities have been implemented and are being used in actual design projects.

2 INTRODUCTION

Model Based Reasoning is a powerful tool used to design and analyze systems composed of numerous interactive, interrelated subsystems. The development of these complex systems requires the following basic steps: specification, modeling, and physical implementation. The specification itself covers three description levels: behavior, logic and structure. The ease with which knowledge is utilized depends on the representation scheme of both the knowledge and the design data. For example, a query to retrieve information might be a one line statement or might entail making several queries, each of which depends on interpreting the results of the previous query. Finding all possible modules active during a particular time sequence might require searching the entire design or might be a single table-lookup query. This paper describes a knowledge representation scheme that can be used at both the specification and model levels. Reasoning from models may cut across any of the levels mentioned and combine information of different models.

One difficulty in knowledge representation is caused by the dynamic nature of the information. The models evolve with the addition and deletion of information, the enhancement of capabilities in the artifact being designed, and the development of better model concepts. Also, the relationships between the models and the knowledge associated with them may evolve.

In this paper, we will address how our system of knowledge representation supports the following aspects of modelling:

- Synthesis of the model from behavioral specification
- Interactions between models
- Evolving models
- Representation of knowledge from diverse domains

The knowledge representation must be capable of expressing the behavior and structure of the system or artifact being designed. These may be described in terms of models which in turn are composed of many different kinds of information, such as equations, constraints, and algorithms. As this information grows and changes it is important to track the evolution. Work done early in a project frequently cannot be used when one returns to that early design, made 6 months ago. Without the models upon which the design decisions were made the work may need to be duplicated using the current models. It would be a lot simpler if the state of the models for each design were kept and the changes were easily accessible. Few people remember whether there were minor or major changes and when they've been made. This is especially true when dealing with a complex system which many designers.

While this form of Model Based Reasoning (MBR) will support many distinct design domains, disparate domains will be used here to illustrate the concepts. The first involves a knowledge-based system, VEHICLES [7] and [7], developed at the Aerospace Corp., that supports the conceptual design of space systems. There are several models for the spacecraft subsystems and their environments (e.g. payload, communications, launch, thermal, etc.). Integrating multiple models that are developed by different people who have focused on different aspects of the design, often at different levels as well, is quite challenging. The knowledge representation scheme we have developed makes the problem tenable.

The second domain involves high level synthesis of VLSI circuits from behavioral level specifications. The knowledge model used in KNOWledge MANager (KNOWMAN) (, art of the Advanced Design AutoMation (ADAM) system developed at the University of Southern California [7] and [7] and [7]) must support many types of knowledge used in the various stages of automated synthesis of VLSI circuits from behavioral descriptions. KNOWMAN consists of a representation methodology for handling the design knowledge necessary for VLSI Synthesis, the implementation of the knowledge representation schema using an object oriented database model, and a set of Prolog expert systems that utilize the data in the database.

The important aspect of this paper is not the specific implementations, but the general application of our proposed methodology to these and other implementations. In Section 2, the basic scheme for knowledge representation will be outlined, and the major concepts associated with the proposed methodology will be stated. Section 3 will present an overview of the knowledge classification structure used in our design systems. Section 4 will summarize and present the status of our work and discuss related problems we are currently working on.

3 REPRESENTATION SCHEME

Our scheme entails

- pooling knowledge types used in models
- tracking models for documentation and historical reference of designs
- design history trace
- explicit representation of the artifact structure
- explicit representation of the knowledge structure, labels
- separation of data and knowledge from processing

In many problem domains, complex representational schemes are necessary due to the complexity of the artifacts being designed [?] and [?]. The inference engine, which uses the domain knowledge, also requires knowledge about how and when to apply that knowledge. The representation scheme includes

- models
- pools, or types, of knowledge
- procedural knowledge
- planning knowledge and
- meta knowledge.

Knowledge is grouped into categories, or pools, which contain all instances of a given category, such as equations or rules. Individual elements from pools of knowledge can be associated with each model. For example, in the design of a spacecraft subsystem, the same equation relating frequency to wavelength for electromagnetic radiation can be associated with the model for an infrared detector payload as well as with the communications subsystem model.

Building complex systems from specifications, which may be incomplete or even inconsistent, requires flexible, extensible models and the capability to utilize the knowledge associated with those models. A particular model and the knowledge associated with it may become obsolete as new data becomes available. New capabilities might be added to the system under design, requiring changes to the model. For example, in VEHICLES, if a circular orbit is selected the orbit model will automatically delete all information (equations, rules, and design configuration values) associated with elliptical orbits. Organizing the knowledge into 'pools of knowledge', such as a pool of equations or a pool of constraints, facilitates the addition of new knowledge categories as well as new instances within a given pool, or type, of knowledge.

To ensure consistency and maintain credibility we have chosen to explicitly represent the structure and associated knowledge of the artifact being designed, the models and knowledge about the design, and the representation scheme itself. Constraints and limitations about the models as part of the knowledge representation are also explicitly represented. For instance, each equation, in the pool of equations may have

- a source (person or reference),
- the time and date it was added,
- annotations (user readable),
- the assumptions upon which it is based and
- the conditions or limits for automatic validity checking.

A subset of equations (or rules, routines, tables, etc.) from the (appropriate) pool is selected for each model. The model itself may include information similar to that for the equations. This makes it possible to keep track of past designs, even when the models have changed. With this peripheral information, we can retrieve the exact model used at any point in a design. This is important for understanding why decisions were made. It might have been due to incompleteness of the models used. This also helps to identify how subsequent enhancements to the models may alter decisions and designs or what technology breakthroughs are needed to meet requirements. For example, in designing a phased array radar many transmit/receive modules are needed to provide sufficient power. A nominal value of 0.3 watts per module would take over 13,000 modules to supply 4000 watts. However, if we could only "afford" the weight of 8000 modules, we would need a technology improvement in modules for them to supply 0.5 watts (hopefully at the same weight).

It is possible to enhance, modify, restructure, remove or prune existing knowledge. Also, the models and associated knowledge can include redundant information in the form of different perspectives or different representations by linking associated concepts. Knowledge that is linked together cannot be automatically removed. An item which is replaced by an equivalent item will automatically have all links updated.

A useful feature in design is traceability. Tracing results makes it possible to identify how each result was derived. This design history provides the source of the data, which could be an equation, a rule, or a designer and the date the result was acquired. This automatically makes it possible to reconstruct the design path, the knowledge and the sequence of its application, which led to earlier design concepts, as well as to the current design concept.

Labels are assigned to the various procedures, facts and models. These labels are made as specific as possible to ensure that rich semantic content will be evident in the state of the program execution, or the state of the knowledge base. The traceability of results is facilitated when the data source is pgm_photon_flux or routine_weight_growth. It also makes the results more credible, because the user untrained in AI techniques can immediately see what factors (e.g. equations, routines) were used in the design.

Frequently, new models are incorporated that only partially overlap with the existing models. This is difficult since one model cannot simply be replaced with another. One type of knowledge that has been quite helpful in facilitating the combination of models is the explicit representation of the linked parameters, i.e. the parameters that are used in a particular model, but supplied from other models, routines, or tables. For instance, the total weight of the spacecraft relies on the individual subsystem weights. Replacement of a subsystem model must still provide the total weight of that subsystem.

Most of the integration of models is done by hand. As our systems continue to be used in additional design projects we are investigating what information would be needed to automate a greater portion of the process. Verifying the consistency and completeness of the knowledge is one aspect currently being worked on. It is already possible to identify conflicts when equations are being solved. A symbolic equation solver will flag problems when values are inconsistent.

The interface, between the design program and the knowledge base, can itself be modified if the knowledge base changes since it is just another type of knowledge. Thus, changes to the knowledge base are accompanied by changes to the interface portion of the knowledge base. The interface knowledge can be used to specify which portions of the updated knowledge base are applicable, thus allowing the proper knowledge from all domains to be utilized. This form of meta knowledge is necessary with a rapidly evolving model or knowledge base [?]. However, it makes the verification of knowledge base correctness, completeness and consistency much more difficult, since there is this meta knowledge which must also be verified since it is susceptible problems as the knowledge.

4 THE KNOWLEDGE HIERARCHY

Design knowledge, which addresses how to do design, what the design is and the requirements to be met, may be represented in a subtype/supertype hierarchy. KNOWMAN, used for synthesis of VLSI circuits, uses an object oriented database, which makes the hierarchical representation very natural. In VEHICLES, which uses a relational database, the hierarchical structure also *is* used. It provides a means to clearly represent the structure of the artifact being designed. It also makes the organization of complicated and voluminous knowledge and data much clearer and easier to access. Two subtypes of knowledge are declarative and procedural. Both of these contain many subtypes in a rich and tangled hierarchy.

Declarative knowledge includes state knowledge, parameterized knowledge, knowledge about goals and assertions. Most statements of facts, descriptions of relationships between objects, and characteristics of objects are members of this type.

State knowledge is knowledge about all of the aspects of the design space: past, present and potential. This is a complex area of knowledge since there are so many aspects that need to be considered. We have selected several candidate areas for inclusion in this type of knowledge:

design space	state history
design paths	current design state
state-dependent data	design approaches
design domains	design drivers
design styles	design situations
design evaluation	design decisions

Parameterized knowledge is particularly important in MBR since the data contained is used to fine tune the models. Equations, tables, ordered lists, graphic knowledge, definitions and image processing data are all examples of the types of knowledge included in this category. Also included are the ranges and units of the attributes associated with the model.

The hierarchy based on the goals associated with the specification being used to drive the planner includes constraints, goals, lower and upper bounds, best achievable designs and actual designs.

Our work in knowledge classification and interaction is ongoing. Other types of knowledge spans many levels, from primitives to higher level constructs. The classification scheme is not rigid, so that the groupings may vary depending on the context. These categories include:

- Symbols, operators, numerics, strings Equations, expressions, functions
- Tables, lists, sets
- Rules, constraints, requirements, goals, scenarios
- Definitions, annotations Instances, data, constants, enumerated options
- Models, frames, scripts
- Graphics, image processing (postscript, gif, pic)
- Report styles, graph styles, highlight items

Under the domain of procedural knowledge, we see the subtypes of heuristics, algorithms, operators, reasoning, rule usage and planning. In KNOWMAN the procedural knowledge is stored in the form of hierarchical data flow graphs where specific procedures are associated with the nodes, and values flow between the nodes. Declarative knowledge can be associated with the procedures by binding specific Prolog code to a given node based on the current state of the design.

5 SUMMARY AND STATUS

We have presented a representational scheme that makes use of models based on pools of knowledge. The structure of the knowledge is semantically rich. This is apparent in the wealth of types, or pools, and the many levels of knowledge that can be represented. Explicit representation will facilitate checking the consistency and ensuring appropriate applicability of information in the knowledge base.

The separation of the inference engine from the knowledge base makes possible the storage of the data in a form that is easily extensible, easily reorganizable and extremely flexible. It also makes it possible to have work proceed on both in parallel.

Complex interactions between subsystems are handled by explicit representation. This facilitates managing the evolution of the models, but it also makes it evident that there is a need for a consistency checker for the knowledge base to ensure the completeness and consistency of the knowledge.

The non-monotonic nature of the evolving models necessitates including *what* and *when* changes occur. This makes it possible to capture design history, providing traceability, and makes the model state dependence explicit. It also helps maintain credibility with the user.

The methodology described here has been, to a large extent, implemented. We have followed the philosophy of separating the knowledge from the inference engine. This became a necessity as the representation scheme frequently changed as new capabilities are continually being incorporated. This

There are some caveats which should be mentioned. As the generality and power of the MBR system grows the performance, especially in terms of speed, decreases. There is also considerable overhead incurred by checking for validity and consistency. The scope of the knowledge represented, which includes both the knowledge itself and its applicability conditions, requires more storage space and necessitates more work in incorporating new models or system capabilities. We are currently investigating solutions to these problems, and see great potential in MBR.

References

[1] K. Bellman and A. Gillam, "A knowledge-based approach to the conceptual design of space systems," in *Proceedings of the 1988 Eastern MultiConference*, pp. 23-27, Mar. 1988.

[2] A. Gillam, "A knowledge-based approach to planning the design of space systems," in *Proceedings of the 1989 Eastern MultiConference*, 1989.

[3] B. Afsarmanesh, *The 3DIS, An Extensible Object Oriented Framework for Information Management*. PhD thesis, USC, 1985.

[4] D. Knapp, *A Planning Model of the Design Process*. PhD thesis, USC, 1986.

[5] D. Knapp and A. Parker, "A unified representation for design information," in *Proceedings of the 1985 Conference on Hardware Description Languages, IFIP*, 1985.

[6] J. Doyle, "A truth maintenance system," *AI Journal*, vol. 12, no. 3, 1979.

[7] E. Shortliffe, *Computer-Based Medical Consultations: MYCIN*. Elsevier, NY, 1976.

[8] R. Davis, "Applications of meta-level knowledge to the construction, maintenance, and use of large knowledge bases," in *HPP Memo and AI Memo*, (Stanford University), 1976.

TELEPRESENCE FOR SPACE: THE STATE OF THE CONCEPT

Randy L. Smith, Douglas J. Gillan, and Mark A. Stuart
Lockheed Engineering and Sciences Company
Houston, Texas

ABSTRACT

The purpose of this paper is to examine the concept of telepresence critically. To accomplish this goal, first, the assumptions that underlie telepresence and its applications will be examined and, second, the issues raised by that examination will be discussed. Also, these assumptions and issues will be used as a means of shifting the focus in telepresence from development to user-based research. The most basic assumption of telepresence is that the information being provided to the human must be displayed in a natural fashion, i.e., the information should be displayed to the same human sensory modalities, and in the same fashion, as if the person where actually at the remote site. A further fundamental assumption for the functional use of telepresence is that a sense of being present in the work environment will produce superior performance. In other words, that sense of "being there" would allow the human operator of a distant machine to take greater advantage of his or her considerable perceptual, cognitive, and motor capabilities in the performance of a task than would more limited task-related feedback. Finally, a third fundamental assumption of functional telepresence is that the distant machine under the operator's control must substantially resemble a human in dexterity.

I. INTRODUCTION

Telepresence is the concept that sensory information from a remote manipulator is of such quality and quantity that a person would actually feel that he or she is at the remote location (Sheridan, 1987). Space operations have been touted as a potential use for telepresence due to the inherent dangers of the operating environment. The purpose of this paper is to examine the concept of telepresence critically. To accomplish this goal, this paper will, first, examine the assumptions that underlie telepresence and its applications and, second, discuss the issues raised by that examination. These assumptions and issues will then be used as a means of shifting the focus in telepresence from development to user-based research.

This discussion will begin by defining some key terms. A robot is a mechanical manipulator arm under direct computer control and most often used in manufacturing. A teleoperator is a mechanical manipulator arm under direct real-time control by the human operator, and a telerobot is a device that spans the capabilities of both systems. It can be controlled directly by the human; however, it has autonomous capabilities that can be exercised, where the human's role changes to that of a systems monitor or supervisor.

Telepresence is of importance in the performance of remote manipulation tasks that take place undersea, in the nuclear industry, and in space due to the significant effects of workstation design on systems performance. NASA is working on several manipulator systems that could require telepresence capabilities. Much of the work on NASA's Space Station Freedom is being planned to be performed by telerobotics. A brief run-down on the participating international space agencies and their respective manipulator systems will give an indication of the scope of planning.

- Canadian Space Agency: Space Station Remote Manipulator System (SSRMS), and Special Purpose Dextrous Manipulator (SPDM);

- NASA: Flight Telerobotic Servicer (FTS), Crew and Equipment Retrieval System (CERS), and Astronaut Positioning System (APS);

- NASDA (Japan): JEM Remote Manipulator System, and JEM Small Fine Arm.

II. ASSUMPTIONS

Basic Assumptions

Three basic assumptions underlie the concept of telepresence. (1) The information being provided to the human must be displayed in a natural fashion, i.e., the information should be displayed to the same human sensory modalities, and in the same fashion, as if the person where actually at the remote site. (2) A sense of being present in the work environment will produce superior performance. In other words, that sense of "being there" would allow the human operator of a distant machine to take greater advantage of his or her considerable perceptual, cognitive, and motor capabilities in the performance of a task than would more limited task-related feedback. (3) The distant machine under the operator's control must substantially resemble a human in dexterity.

Corollary Propositions

These three basic assumptions produce the corollary propositions listed below:

(1) sensory information should not be perturbed or transformed due to its deleterious effects on the quality of performance,

(2) sensory information will need to be high resolution and high fidelity,

(3) humans will need multi-sensory information from a work environment,

(4) users will require both task-relevant and task-irrelevant sensory information, depending on what is needed to provide a sense of "being there,"

(5) humans will need several different types of control devices for making inputs to the telerobotic system(s), often making inputs simultaneously.

III. ANALYSES

The fundamental assumptions and their corollary propositions indicate a need for analysis of telepresence -- by experimental investigation, task analyses, trade studies, technology evaluations, and cost-benefit analyses before telepresence can be demonstrated to be a useful concept. At present, particularly strong support exists for the proposition that information should not be perturbed or transformed. Sensory modalities that the human normally uses might well allow the human to capitalize on a lifetime of experience in interpreting sensations, thereby reducing his or her mental workload. Research by K. U. Smith and others document the effects on performance of presenting an operator with "unnatural" sensory information. Smith, Smith, Stuart, Smith, and Smith (1989); Smith (1988); Stuart and Smith (1989); and Chandlee, Smith, and Wheelwright (1988) discuss the sometimes deleterious effects that video images, illumination, and perturbed sensory feedback have on operator performance of a remote manipulator.

Research has documented the adverse effects that spatial and temporal perturbations have on direct manipulation tasks. Smith and Smith (1962) found that video displacements produced emotional and organic disturbances ranging from "minor emotional disturbances and frustration, through dizziness, giddiness, faintness, to nausea and illness" (p. 108). Small angular spatial displacements were effectively adapted to; however, adaptation to larger displacements was much less successful. Adaptation varied with the type of motion involved and was not significantly affected by a lack of knowledge about the conditions of displacement. The adaptation was more dependent on correspondence between visual feedback and motion.

Adaptation to displacement was very specific to a situation and little transfer of learning occurred between different situations, although specific skills were retained for up to two years. Smith and Smith (1987) summarized the adaptation research in a theory of behavior, which they call the neurogeometric theory, in which "space perception and visually controlled movement are learned, the nature and degree of learning are determined by the nature and degree of spatial compliance between muscular control and sensory input" (p. 268).

User-Based Research Issues

The rest of the paper will describe a program of analyses that need to be undertaken prior to future development in telepresence systems. The major issues can be broken into the following categories, derived from the assumptions and corollaries:

(1) How great must the performance advantage be in terms of a cost/benefit ratio, given a high cost of providing "natural" sensory information (in development costs as well as use of bandwidth resources in the operating environment). Wilson and MacDonald (1986) state "the first problem of telepresence -- it is often impossible or impractical to provide it."

(2) How many senses need to be used for the person to acquire the illusion of actually being at the remote site?

(3) How high does the fidelity of the information need to be, especially given restrictions on human information processing resources? One major problem with the idea of telepresence is human sensory and cognitive processes, such as adaptation, habituation and selective attention, serve to filter much of the sensory stimulation that a person receives. A second problem is that much of the information that is provided to the operator may be distracting.

(4) What types of information and under what circumstances is it inadvisable to feed back information? Wilson and MacDonald state "there are obviously conditions in some work environments that it would be fatally inappropriate to feed back."

(5) What are the effects of cross modality feedback, i.e., the effects of feeding back sensory information to sensory modalities different from those normally used, such as feeding force information back visually rather than tactilely?

(6) What are the effects of perturbed sensory feedback?

Technical Research Areas

The main technical areas that are in need of evaluation and research, can be categorized as the following based on work by Sheridan (1987): (1) telesensing, (2) teleactuating, (3) computer-aiding in supervisory control, and (4) theory and analysis of human-telerobot interaction. Each of these four areas will be described in detail below.

Telesensing. Telesensing concerns sensing the environment of the telerobot at a distance. It encompasses visual feedback, resolved force information, touch, kinesthesis, proprioception, and proximity sensing. Vision feedback issues in need of investigation are the numbers and positions of telerobot cameras, resolutions of video systems, other parameters of video systems (such as color vs. black & white), digital image processing as an aid to the human operator, stereo vs. monocular feedback, head-coupled displays, and human-telerobot perception research (determination of perceptual process involved in detection, discrimination and recognition).

Force feedback concerns sensing what the human body's joint, muscle and tendon receptors do to determine the net force and torque acting on the hand. For a manipulator this force information is measured at the remote scene and fed back to the human. Areas for research include cross modality feedback, i.e., examining the effects of feeding back force information to the tactile, visual, or auditory mode. Force feedback with time de-

lays is a separate issue in need of investigation as this effects the stability of the control system and the usability of the information for the human. Touch feedback concerns the pressure sense of the skin. Cross modality feedback of force is a possibility as well. Other research areas include grasp integrity, object recognition, object orientation/ alignment, and surface/edge detection.

Kinesthesis, the sense of motion, and proprioception, the awareness of where in space one's limbs are, may be particularly a problem in microgravity environments. For telerobotics, visual information is the primary feedback concerning manipulator arm positions and orientations; however, for human beings much information about arm positions is received from the proprioceptive sense. Research areas to address manipulator positional awareness include: multiple camera views; wide field of view cameras from a vantage point which includes the arm base; and computer-generated images of various kinds. Humans do not have the ability of proximity sensing via optical or laser ranging, but it may be useful for certain telerobotic tasks.

Teleactuating. The types and effects of various controllers on humans and systems performance needs to be evaluated. (See Smith and Gillan (1987) for a discussion of various mental models telerobot operators may possess dependent on controller type.) Additional issues concern commonality of devices across systems. To reduce equipment cost and weight, and crew training, the number of control devices should be minimized. Other issues concern anthropometrics of control devices (how they physically fit the entire population of users) and control modes (rate, position, hybrids). Two-armed interaction effects include situations when multiple manipulator arms are used in a coordinated or simultaneous fashion to accomplish a task. Multiple-person cooperative control is a problem similar to multiple-arm coordination where more than one person is performing a single task.

Adjustable impedance of the master controller and/or slave arm concerns the area of matching and mapping force between the ends of the control loop. Interchangeable end effector tools need evaluation, as do task-resolved manipulation. Task-resolved manipulation concerns performing standard or preprogrammed operations relative to the surface of an environmental object, i.e., sensing the surface in the process of manipulation and continually performing coordinate transformations to update the axes with respect to which operations are being performed (Sheridan, 1987). This is an extension of end-point resolution. Cross modality control inputs concern inputting commands to a telerobot subsystem via a different mode than would be performed by a human operating their own human subsystem. An example is using head-slaved camera control to control camera pan/tilt (which appears more natural) versus using voice commanded systems to control camera pan/tilt (inputting with an unnatural mode).

Computer Aiding in Supervisory Control. Supervisory control can be used when the human is functioning as more of a system monitor or supervisor and the telerobot is performing tasks autonomously. Supervisory control could be used for crew training. Additional research areas include the following: state measurement, estimation, and display; and aids for failure detection, identification, and emergency response.

Theory and analysis of human-telerobot interaction. An overall theory of how humans interact with advanced teleoperated systems may also be helpful so that research in different areas can be addressed and the integrated effects of various control and display devices can be evaluated. Behavioral cybernetics is proposed to be such a theory. It concerns the communication and control parameters of automated systems and can be used as a basis for display/control performance models for human operators. Smith *et al.*, (1989) discuss the phenomenon of perturbed sensory feedback as a potentially serious obstacle to optimal performance and safety of telerobotic tasks. This theory would aid the synthesis of answers to design problems by allowing us to "go beyond our data." Rouse (1987) suggests human factors professionals be prepared to extrapolate from specific studies to support design decisions

that would otherwise be made *ad hoc*. This theory could serve as the basis for a method of synthesis of data to design problems based on an understanding of the interplay between behavioral, organizational, and technological factors.

IV. CONCLUSIONS

Prior discussions and work on the development of telepresence prototypes have been valuable in that they have allowed us to identify a wide variety of analyses that should be performed prior to developing an advanced telepresence system. See Figure 1 for graphical representation of issues, corollary propositions, and assumptions concerning the state of the concept of telepresence. Resources should now be shifted towards the types of analyses described in this paper. Even if the analyses ultimately indicate that telepresence is not a useful or a realizable technology, the information derived will be usable in the design of the human-telerobot interface in general. However, continuing down the path of developing telepresence systems without performing adequate user-based research runs two risks: (1) money will be spent that will never lead to a practical system, and (2) developing a technology that human operators do not want and do not need to perform their tasks.

V. ACKNOWLEDGEMENTS

Support for this work was provided by the National Aeronautics and Space Administration through Contract NAS9-17900 to Lockheed Engineering and Sciences Company.

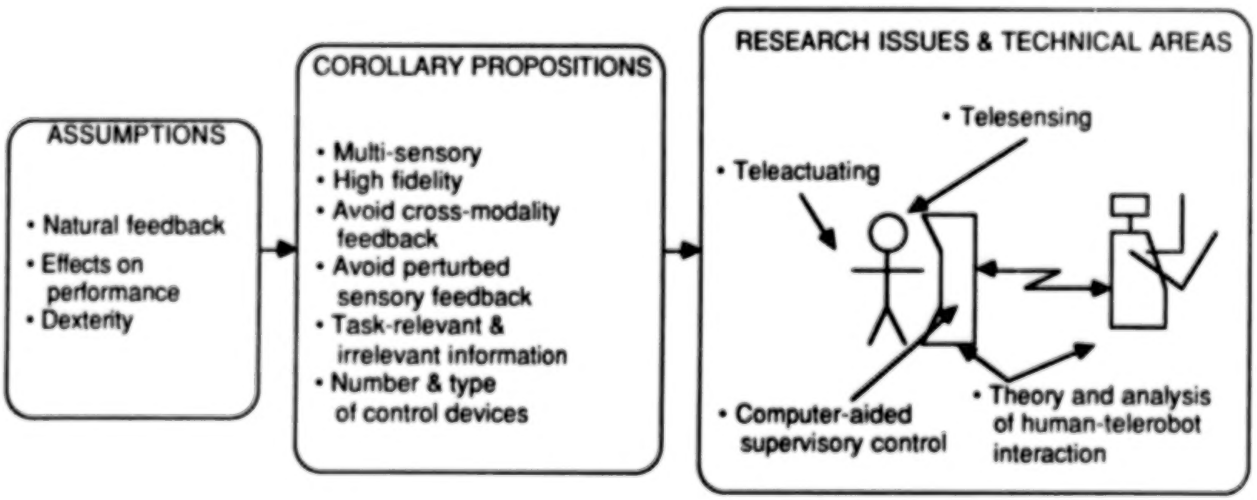


Figure 1. Assumptions, corollary propositions, and research issues & technical areas concerning the state of the concept of telepresence.

VI. REFERENCES

Chandlee, G. O., Smith, R. L., and Wheelwright, C. D., "Illumination Requirements for Operating a Space Remote Manipulator," in W. Karwowski (Ed.), ERGONOMICS OF HYBRID AUTOMATED SYSTEMS I, Amsterdam: Elsevier, 1988, pp. 241-248.

Mohr, G. C., "Robotic Telepresence," PROCEEDINGS OF THE HUMAN FACTORS SOCIETY 30TH ANNUAL MEETING, 1986, Santa Monica, California.

Rouse, W. B., "Much Ado About Data," HUMAN FACTORS SOCIETY BULLETIN, 30, (9), 1987, pp. 1-3.

Sheridan, T., "Teleoperation, Telepresence, and Telerobotics: Research Needs for Space," in T. Sheridan, D. Kruser, and S. Deutsch (Eds.), HUMAN FACTORS IN AUTOMATED AND ROBOTIC SPACE SYSTEMS: PROCEEDINGS OF A SYMPOSIUM, Washington, DC, 1987, pp. 279-291.

Smith, K. U., and Smith, W. M., PERCEPTION AND MOTION: AN ANALYSIS OF SPACE-STRUCTURED BEHAVIOR, W. B. Saunders Company, Philadelphia, Pennsylvania, 1962.

Smith, R. L., "Human Visual Requirements for Control and Monitoring of a Space Telerobot," in W. Karwowski (Ed.), ERGONOMICS OF HYBRID AUTOMATED SYSTEMS I, Amsterdam, 1988, pp. 233-240.

Smith, R. L., and Gillan, D. J., "Human Telerobot Interactions: Information, Control, and Mental Models," PROCEEDINGS OF THE HUMAN FACTORS SOCIETY 31ST ANNUAL MEETING, Santa Monica, California, 1987.

Smith, T. J., and Smith, K. U., "Feedback-Control Mechanisms of Human Behavior," in G. Salvendy (Ed.), HANDBOOK OF HUMAN FACTORS, John Wiley and Sons, New York, 1987, pp.251-293.

Smith, T. J., Smith, R. L., Stuart, M. A., Smith, S. T., and Smith, K. U., "Interactive Performance in Space - The Role of Perturbed Sensory Feedback," PROCEEDINGS OF THE HUMAN COMPUTER INTERACTION 1989 INTERNATIONAL CONFERENCE, Boston, Massachusetts, 1989.

Stuart, M. A., and Smith, R. L., "SPATIALLY DISPLACED VISUAL FEEDBACK AND THE PERFORMANCE OF FTS-LIKE TASKS," NASA Tech. Report JSC-23547, Houston, Texas, 1989.

Wilson, J. and MacDonald, D., "Telepresence - Goal or Byproduct of Remote System," PROCEEDINGS OF ROBOTS 10 CONFERENCE, Chicago, Illinois, 1986.

TELEPRESENCE AND SPACE STATION FREEDOM WORKSTATION OPERATIONS

Dean G. Jensen
NASA, Johnson Space Center

Susan C. Adam, James H. Stramler, and Robert P. Wilmington
Lockheed Engineering and Sciences Company

ABSTRACT

The Space Station Freedom workstation system is a distributed network of computer based workstations that provides the man-machine interfaces for controlling space station systems. This includes control of external manipulator, robotic and free flyer devices by crewmembers in the space station's pressurized shirt-sleeve environment. These remotely controlled devices help minimize the requirement for costly crew EVA time for such tasks as station assembly and payload support. Direct window views may be used for controlling some of the systems, but many activities will be remote or require levels of detail not possible by direct observation. Since controlling remote devices becomes more difficult when direct views are inadequate or unavailable, many performance enhancing techniques have been considered for representing information about remote activities to the operator. This paper describes the telepresence techniques under consideration to support operations and training. This includes video enhancements (e.g. graphic and text overlays and stereo viewing), machine vision systems, remote activity animation, and force reflection representation.

INTRODUCTION

The space station workstations will be distant from many of the devices they control. This is unlike the Shuttle manipulator workstation which is an optimally located workstation dedicated to providing an operator the necessary controls and displays for the Shuttle remote manipulator system. Because this workstation is close to the manipulator system, most tasks controlled from it can be observed directly through windows in the aft flight deck. Direct observation of remotely controlled equipment makes a task easier and assists an operator in maintaining an overall orientation even when camera views are available. Although some devices controlled from the space station workstations will be observed through windows, many tasks done by operators at space station workstations will involve controlling devices for which direct views are either inadequate or unavailable because of distances or obstructions. As can be seen in Figure 1, direct observation of remote truss activities controlled from the forward node workstations will be difficult. Operators at these workstations will require additional information about external events to control the devices safely and productively. This paper briefly describes the space station workstation system, the devices that will be controlled from the workstations, and the different techniques that have been considered for providing space station operators external event information (or telepresence).

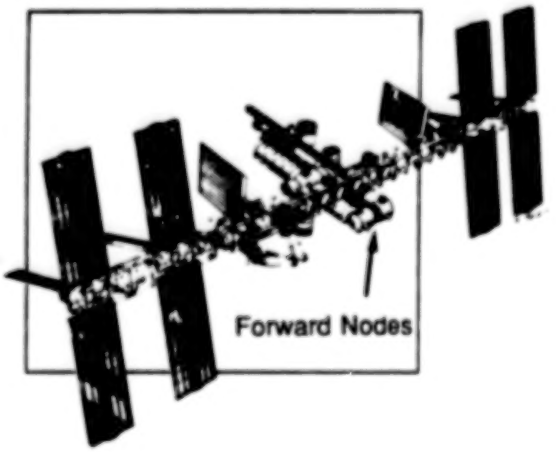


Figure 1. Space Station Freedom

SPACE STATION WORKSTATION SYSTEM

The workstations in the space station will be distributed throughout the habitable areas. A key characteristic of this system is that access to systems control is accomplished through the distributed computer system. In this respect each of the individual workstations is like a terminal on a large centralized system. However, each workstation will contain local processing abilities so that unlike a centralized system, activities at each workstation will have the capacity for independent activity. By using the distributed computer system each workstation will have the capacity to access control of space station systems (such as the Electrical Power, Data Management, Thermal Control, and Guidance,

Navigation, and Control Systems) and many of the workstations will be able to control the devices outside the pressurized volumes.

There are three main categories of workstations: Cupola, fixed, and portable. The Cupola workstations, which are primarily configured to control devices outside the pressurized volumes (robots and free flyers), provide direct views of external activities through windows. One cupola is positioned on the nadir of one forward node and the other is positioned on the zenith of the other forward node (see Figure 2). Each cupola will provide viewing for the upper or lower (±Z axes) areas of the space station. These cupolas are small dome shaped structures that contain windows all around the sides (±X and ±Y axes) and in the top (or bottom for the nadir cupola) and are large enough for two side-by-side crewmembers to work. The workstations in the cupolas will have access to systems control through the space station distributed computer system but are expected to be used primarily for controlling external space station devices that require direct observation. The cupola displays and controls are reconfigurable within the cupola so that operations can be performed in either the forward, aft, starboard, or port directions. They differ from the fixed workstations in that they are reconfigurable and they differ from the portables in that they cannot be removed and used remotely from the cupolas.

The second category of workstation is the fixed workstation. These are workstations that fit in the standard 40" wide x 80" high (approx.) space station racks. Within the fixed category are two types: Command & control and standard workstations. The command & control workstations are required to have the capabilities of the cupola as well as the capabilities of any other fixed workstation. The command and control workstations are expected to be used for general station flight management and for controlling external devices which require video views rather than direct views (e.g. controlling a dexterous manipulator at the end of the space station truss).

Each of the space station laboratory modules will have a fixed workstation that will be used primarily to support the laboratory activities. These include the U. S. Laboratory Module, the ESA APM (Attached Pressurized Module), and the NASDA JEM (Japanese Experiment Module). The aft end of the JEM will also house a fixed workstation for operating the JEM's external robots. This workstation will not be part of the distributed workstation system. The devices that it will control can only be controlled from this workstation and this workstation will not control distributed systems. It services the adjacent external exposed facility that is observed through a window in the aft bulkhead of the JEM.

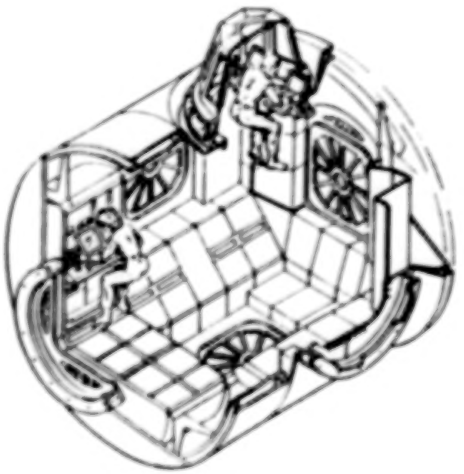


Figure 2. Forward Node Command & Control and Cupola Workstation Concepts

The third category of workstation is the portable. The portable workstations may be the size of large "lap-tops" and will provide access to the distributed computer system to support such activities as crew health care, crew operations, airlock and hyperbaric airlock activities, logistics, and payload support in areas remote from fixed workstations. They will also be used to supplement fixed workstation operations when extra processing, controls, or displays are required.

The workstations involved in operations requiring telepresence are primarily the cupola, command & control, and the JEM's workstation for controlling external robots.

FREE FLYERS AND ROBOTS

There are three categories of devices controlled from the cupola, command & control, and the JEM workstations that will use telepresence. These are free flyers, large manipulators, and dexterous manipulators.

The free flyers include the ESA MTFF (Man Tended Free Flyer), the CERS (Crew and Equipment Retrieval System), and the OMV (Orbital Maneuvering Vehicle). The MTFF will operate as a space laboratory with a number of experiments. This will be mostly unattended or automatically attended by on-board tele-control or ground tele-supervision. Periodically the MTFF will dock at the space station for servicing and parts replacement. The CERS is

required to retrieve incapacitated EVA crewmembers or equipment separated from the space station. The OMV is required to posit or retrieve payloads (ranging from 3,500 to 7,500 lbs.) in other orbits and then return to the space station.

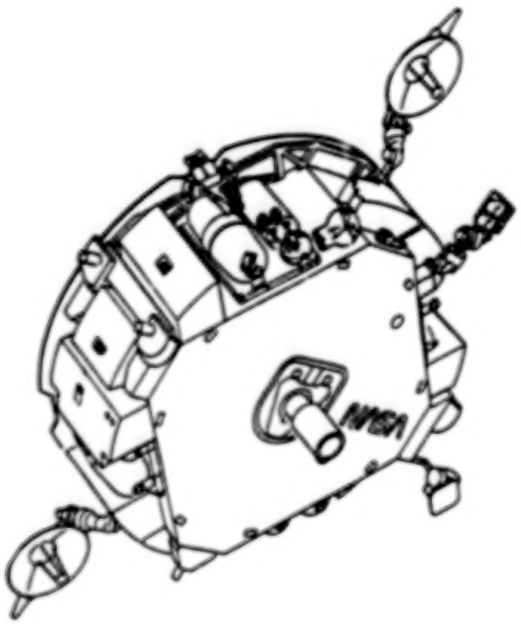


Figure 3. Orbital Maneuvering Vehicle

The large manipulators include the SSRMS (Space Station Remote Manipulator System, as shown in Figure 3) and the JEM RMS (Remote Manipulator System). These arms will be used much like the Shuttle RMS for moving grappled objects from one location to another. The objects moved will range from modules delivered by the Shuttle that will form the habitable volumes to the dexterous robots that are moved to and from work locations. The SSRMS (as well as dexterous manipulators, payloads, and orbital replacement units) can be moved about the station truss by the mobile transporter. The JEM RMS is dedicated to JEM exposed facility operations (as is the JEM small fine arm).

The dexterous manipulators include the FTS (Flight Telerobotic Servicer), the SPDM (Special Purpose Dexterous Manipulator), and the JEM small fine arm (the FTS and JEM small fine arm are shown in Figure 4). These are robots that will be used for assembly, maintenance, and payload tasks of the type that currently require an EVA astronaut. These have arms that are approximately six feet long with end effectors that accept different tools for removing, installing, manipulating, and servicing equipment. These are similar to robots currently used in environments that are hostile to humans such as in undersea and nuclear industries.

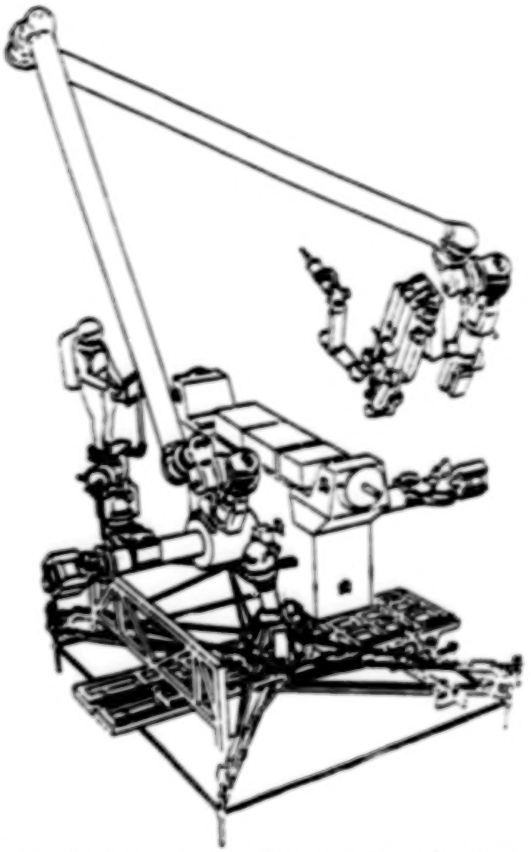


Figure 4. Space Station RMS on the Mobile Transporter shown with the SPDM Attached

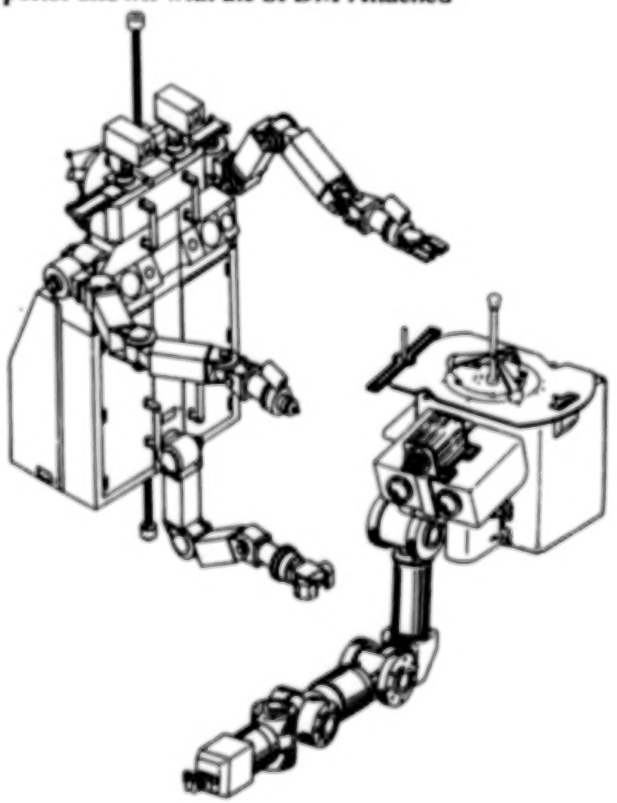


Figure 5. FTS (Top) and Japanese Small Fine Arm (Bottom) Dexterous Manipulators

TELEPRESENCE TECHNIQUES

The telepresence techniques that have been considered for operating the free flyers and robots from space station workstations include video systems and enhancements, machine vision systems, animation, and force reflection.

VIDEO

The most commonly used telepresence technique is video. This often involves the use of multiple video views to provide orthogonal views of the task area. The proper use of lighting to assist depth perception (e.g. with shadows and light patterns) can also facilitate tasks that must be accomplished from camera views. The SSRMS and the JEM RMS will have cameras at the joints as well as at the grapples for viewing operations and targets. The dexterous manipulators will have cameras at the base of the arms (or at the body of the robots) as well as at the end effectors for viewing the manipulations. The space station truss will also have cameras and lights positioned along it to provide general viewing to support these operations.

Enhancement techniques have been developed for video systems. These include stereo viewing and graphics and text overlays on live video. Stereo viewing allows an operator at a workstation to perceive a three-dimensional image which facilitates depth perception in remotely viewed tasks[3]. Some of the systems require the use of glasses much like those used in the 3-D movies to achieve the effect. Other systems are able to achieve the effect without requiring the operator wear any devices but this requires precise eye positioning and small movements of the head may result in the loss of the three-dimensional perception. Another type of three-dimensional display is the head or helmet mounted display which can have individual displays for each eye[5]. The three dimensional systems are only effective for remotely observed events that are within about six feet of the cameras. Consequently, stereo vision systems are considered primarily for dexterous manipulators.

Graphics and text overlays on video are used to aid operators in alignment tasks. For example, the OMV must be grappled by the SSRMS as it approaches the space station and then it must be docked. This requires two operators to work cooperatively. One operator flies the OMV while the other operates the SSRMS. Simulations have indicated that graphic circles overlayed on the live video of the mating structures have been useful in assisting the operator in aligning the devices as they are brought together. Text presented overlayed on the video can provide information on how far apart the objects are and how quickly they are coming together.

MACHINE VISION SYSTEMS

Machine or automatic vision systems use optical or laser systems that respond to targets to calculate positions and rates. Optical systems use high contrast targets arranged in a pattern on an object that is in an unknown orientation at an unknown distance. The pattern of light that is reflected is processed to give information about the object's orientation and distance[6]. This information can then be displayed at a workstation in either graphic or text form to assist the operator. This has been considered as an augmentation to video to assist an operator in docking procedures or to provide the data for automated orientation sequences performed by a robot.

Laser systems have the advantage of not being sensitive to the dramatic lighting and shadow conditions found in space. The SPDM is expected to contain a computer vision processor and a laser scanning sensor that will produce high-resolution three-dimensional maps of the worksite[7]. Current research is addressing such questions as using model-based camera and laser-scanner machine vision techniques that may be appropriate for space applications[1].

ANIMATION

Animation involves representing external events in the form of graphic presentations. These can range from two points in the screen that indicate relative distances and closing rates to near-fidelity representations of three dimensional objects.

In its simplest form, animation may be used to present information acquired from a machine vision system in graphic rather than text format. Rather than seeing numbers presented on the screen that represent the distances and rates between two closing objects an operator could see two simple graphic forms coming together. The movements between the forms on the screen represent the distances, directions of movement, and relative rates between the real objects being manipulated outside the space station.

In its most complex form, animations could be derived from the models of the space station kept in databases. One technique being considered for collision detection and avoidance is the use of a

database that contains information about all the space station components. This database would contain all the data about dimensions and locations and would be used to generate a three-dimensional dynamic graphical model of the space station. Any movement by an RMS, free flyer, or dexterous manipulator could then be modeled with the rest of the space station and provide information about conflicts that could result in collisions. This model would then be used to restrict movements of devices that would result in collisions.

If this database is developed and used for collision detection and avoidance, it has been suggested that it could also be used to provide operator aids at the workstations. The database could then be used to generate "birds-eye" views from points where there are no cameras. These types of global views have been shown to be useful in preventing disorientation when operators have no direct view of remotely controlled events. It could also be used to generate views that are behind objects or otherwise obscured from the real cameras.

FORCE REFLECTION

Force reflection is giving information to the operator or the system about forces that are experienced by a device. The information may be felt, seen, or heard by the operator. Force reflection may also be part of a closed loop system in a robot to prevent it from exceeding certain forces. In this mode, the operator may receive no force information at all and the robot will be prevented from exerting forces that will damage equipment. The force reflection that is instrumental in telepresence is the force reflection that provides information to the operator and that is the type that will be described.

Force reflection that is felt (hand-controller force reflection) can allow the operator to "feel" when the end effector has made contact and the amount of force being exerted. This mode, sometimes referred to as bilateral force reflection, is one in which the forces exerted on the device being controlled are reflected or conveyed to the device (e.g. hand controller) that is controlling the robot. In this manner the controller and the robot are tightly coupled so that forces on one are reflected in the other. For master type of controls, this means motions that force movement in the robot also can force movement in the controller and the operator. These types of controllers often have force limitations at the controller to prevent an operator from being injured by accidental events at the end effector.

This type of force reflection offers several advantages. It has been shown to facilitate inexperienced operators or operators that are doing unscheduled or novel dexterous manipulator tasks[4]. It can be useful for providing information about objects that cannot be seen. In a limited fashion, the end effector can reach behind something and "find" a protrusion or a hole by providing limited topological information to the operator. In addition, operators performing representative remote handling tasks using force reflection can have lower error rates, lower peak forces, and more consistent application of forces than without force reflection[2].

There are also several disadvantages. Force reflecting hand controllers used at a space station workstation will require the operator to be restrained against the forces exerted by the hand controllers. Since the operator will be in a weightless environment, even the smallest forces will cause the operator to move. This type of force reflection can also be fatiguing since the operator has to work against the force reflection. Also, typical force reflection systems are more expensive in mass, volume, and individual unit prices than systems without force reflecting capabilities[2].

Aural force reflection provides auditory displays so that information about forces at the end effector are perceived by sound rather than by feel. In this manner, force reflection may be perceived as sound that varies in intensity or pitch in proportion to the forces exerted at the end effectors. Simpler versions of aural force reflection include alarms that sound when certain forces have been exceeded.

Visual force reflection provides visual cues rather than forces that are felt or sounds that are heard. This information may be presented on a computer display as text that indicates forces and force vector information or as graphics with lengths or areas that are displayed in proportion to the forces at the end effector. These displays may use light intensity, color, or on/off status to indicate forces. This form of force feedback has a disadvantage in that the information will not be perceived unless the operator is actively attending to it.

SUMMARY

This paper has given an overview of the space station workstations, the devices the workstations will control, and the telepresence techniques that have been considered to support the control of those devices from the workstations. The telepresence techniques used at space station workstations will

likely involve combinations of some of the concepts described above. Given the variety of the devices controlled from the workstations it is possible that different techniques will be chosen for different operations. The crew may have the option of choosing between techniques for some operations. Facilities that support the development of space station systems are currently evaluating these techniques to determine the appropriate applications for each of the space station devices and their respective tasks.

REFERENCES

1. deFigueiredo, R.J.P., and Kehtarnavaz, N. "Model Based Orientation - Independent 3-D Machine Vision Techniques", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 24, No. 5, September, 1988, pp. 597-607.

2. Draper, J. V., Herndon, J.N., and Moore, W.E. "Implications of Force Reflection for Teleoperation in Space", Presented at the Goddard Conference on Space Applications of Artificial Intelligence and Robotics, Greenbelt, Maryland, May 14, 1987.

3. Kim, W. S., Ellis S.R., Tyler, M.E., Hannaford, B., and Stark, L.W. "Quantitative Evaluation of Perspective and Stereoscopic Displays in Three-Axis Manual Tracking Tasks", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-17, No. 1, January/February, 1987, pp. 61-72.

4. Molino, J. A., and Langley, L.J., "Testing the Feasibility of Using a Teleoperated Robot to Perform Tasks in Nuclear Test Facilities", TUF 88-005, Nuclear Effects Division U.S. Army Material Test and Evaluation, White Sands Missile Range, New Mexico, March, 1988.

5. Stark, L., Tendick, F., Kim, W. and students "Telerobotics: Problems and Research Needs", *IEEE Transactions on Aerospace and Electronic Systems*, 23007, Vol. 24, No. 5, September, 1988, pp.542- 551.

6. Tryggvason, B.V., Pinkney, H.F.L., MacLean, S.G., Garneau, M., Perrratt, C.I., and Aikenhead, B.A., "Mission 41-G Tests Supporting the Development of a Space Vision System", *Canadian Aeronautics and Space Journal*, Vol. 31, No. 3, September, 1985, pp. 256-267.

7. Hughes, R.C., and Hunter, D.G., "The Special Purpose Dexterous Manipulator (SPDM): A Canadian Focus for Automation and Robotics on the Space Station", Presented at the AIAA/NASA First International Symposium on Space Automation and Robotics, Arlington, Virginia, November, 1988.

THE HUMAN FACTORS OF WORKSTATION TELEPRESENCE

Thomas J. Smith*
Human Factors Group
Bureau of Mines
5629 Minnehaha Ave. S.
Minneapolis, MN 55417

Karl U. Smith
Emeritus Professor
Behavioral Cybernetics Laboratory
Univ. of Wisconsin - Madison
1001 Tower Blvd., Lake Wales, FL

ABSTRACT

Human-operated, remotely controlled robots (telerobots) are projected to play a pivotal role in the performance of assembly, maintenance, and servicing manipulation tasks, during construction and operation of the U.S. space station in the next decade. To reap the anticipated benefits of telerobotic systems---increased safety, efficiency, and productivity of task performance in space, accompanied by reduced costs---it is essential that the control requirements for telerobot operation are compliant with control capabilities and limitations of the human operator. The term workstation telepresence has been introduced to describe such human-telerobot compliance, which enables the human operator to effectively project his/her body image and behavioral skills to control of the telerobot itself. This report addresses major human-factors considerations for establishing high fidelity workstation telepresence during human-telerobot operation. Telerobot workstation telepresence is defined by the proficiency and skill with which the operator is able to control sensory feedback from direct interaction with the workstation itself, and from workstation-mediated interaction with the telerobot. Numerous conditions influencing such control have been identified. This raises the question as to what specific factors most critically influence the realization of high fidelity workstation telepresence? The thesis advanced in this report is that perturbations in sensory feedback represent a major source of variability in human performance during interactive telerobot operation. Perturbed sensory feedback research over the past three decades has established that spatial transformations or temporal delays in sensory feedback engender substantial decrements in interactive task performance, which training does not completely overcome. Similar, more recent laboratory studies with remote telerobot manipulators have confirmed in part the earlier findings. The goal of effective and safe interactive telerobot operation therefore may benefit from development of techniques which enable the interactive computer to detect, and compensate for, perturbations in sensory feedback before presentation of such feedback to the operator. A recently developed social cybernetic model of human-computer interaction can be used to guide this approach, based on computer-mediated tracking and control of sensory feedback. The report will conclude by indicating how the social cybernetic model can be employed for evaluating the various modes, patterns, and integrations of interpersonal, team, and human-computer interactions which play a central role in workstation telepresence.

---

*This is not an official Bureau of Mines report. The views expressed are not necessarily those of the U.S. Bureau of Mines.

INTRODUCTION

Automation and robotics (A&R) will play a pivotal role in the development and operation of permanently-manned U.S. extraterrestrial outposts. One such outpost currently under development by NASA is Space Station Freedom, an orbiting space laboratory targeted for assembly in the mid-1990s. Projected A&R applications for the U.S. space station encompass over twenty major projects, including use of interactive teleoperated robots (telerobots) for assembly, servicing, and maintenance functions [1-3]. Telerobots are devices with mechanical arms for grasping and manipulation which are envisioned to play an integral role in space station operation. The devices are projected to have some of the same operational capabilities as an astronaut in a space suit. NASA and Bureau of Mines research also indicates that interactive A&R systems will be employed in mining lunar resources, in support of a lunar base in the 21st century [4-8]. Anticipated benefits of interactive A&R applications in space include a reduced crew size, increased crew productivity, consequent cost savings, and increased safety of the crew, concomitant with use of telerobots instead of crew members to conduct risky extravehicular (EVA) maneuvers [1,2]. For these benefits to be realized, it is essential that the human-factors design and operational characteristics of the human-computer/human-telerobot interface be compatible with the behavioral-physiological performance capabilities and limitations of the human operator. In the case of human operation of telerobots at remote sites (teleoperation), such compliance across the interface is described by the term telepresence, defined by NASA as "a teleoperation situation in which the operator has sufficient cues to simulate sensations that would be experienced in direct manual performance of the operations" [9].

This report provides a conceptual and technical analysis of the major human-factors issues which must be addressed in creating effective workstation telepresence during telerobot operation. A major focus of the analysis concerns the phenomenon of perturbed sensory feedback as a decisive influence on workstation telepresence. The term refers to the introduction of some sort of perturbation, or transformation, in the spatial, temporal, or physical properties of one or more modalities of sensory input to the human operator across the interface. Such perturbations alter the normal closed-loop control mechanisms of cognitive and motor perceptual behavior, and can thereby evoke substantial decrements in human performance during teleoperation.

The principle hypothesis advanced in this report is that perturbations in sensory feedback constitute a major obstacle to the creation of high fidelity workstation telepresence. As a result of

human-factors limitations of interface design, perturbations in sensory feedback arise routinely during human interaction with telerobotic systems. These limitations may be exacerbated by conditions prevailing in space environments. To address this problem, the following sections: (1) provide a human-factors analysis of telerobotic workstation telepresence; (2) outline the adverse effects of perturbed sensory feedback on interactive performance for behavior generally, and for telerobotic operation in particular; and (3) discuss how such effects can be managed operationally, based on use of a social cybernetic model of workstation telepresence to target research needs.

HUMAN-FACTORS ANALYSIS OF WORKSTATION TELEPRESENCE

During human performance with a telerobot via a teleoperation workstation, sensory feedback (visual, tactile, kinesthetic, and/or auditory) to the operator is generated by two sources: direct interaction with the workstation, and workstation-computer mediated interaction with the telerobot. To optimize workstation telepresence, the spatial, temporal, and physical properties of workstation and telerobot sensory feedback, as determined by the human-factors design of the teleoperation system, must be compliant with the sensory feedback control capabilities of the human operator.

This concept is illustrated in Figure 1, using the example of a movement-actuated (master-slave) telerobot manipulator. Effective control of the manipulator depends upon space, time, and force (kinesthetic) compliance between operational feedback from the telerobot slave sector, and motor behavioral control of this sensory feedback by the operator. If human-factors design problems cause spatiotemporal perturbations in sensory feedback, then compliance between sensory feedback and its control is compromised and performance suffers. Evidence from various sources suggests that because of human-factors design problems, both spatial and temporal perturbations in sensory feedback are a pervasive feature of interactive human-computer and telerobotic systems [10-14].

Figure 2 expands upon Figure 1 by schematically illustrating the major human-factors design issues that must be addressed in creating workstation telepresence [11]. Figure 2A shows a telerobotic manipulator under direct control, rather than computer-mediated remote control, by the operator. Integrated postural, transport, manipulative, and tremor movements are employed to control the spatiotemporal properties of manipulator operation, in relation to its master-control, actuator, and slave components (Figure 1). For the control task, the operator relies upon three major sources of sensory feedback: (1) reactive feedback from movements of body segment and limb effectors; (2) instrumental feedback from effector interaction with and movement of the manipulator master-control component; and (3) operational feedback from action of the manipulator slave component and its effect on the environment.

Figure 2B illustrates how these relationships are altered in the case of teleoperation of a remote manipulator. In this case, control of the manipulator is mediated through a master-control computer workstation, whose actual design for space station telerobotic applications is still under development [15,16]. Direct reactive, instrumental, and operational sources of sensory feedback are generated by operator interaction with the workstation, rather than directly with the telerobot. Through display of video images from external cameras, plus tactile, force, and possibly auditory sensors mounted on or near the robot, the workstation provides indirect, computer-mediated reactive, instrumental, and operational sensory feedback information from activity of the telerobotic manipulator. Thus, during remote telerobot manipulation, the operator must simultaneously control sensory feedback from two major modes of interaction, namely direct interaction with the workstation plus workstation-mediated interaction with the telerobot manipulator. Collectively, this combination of direct plus indirect, computer-mediated feedback information may be referred to as telepresence sensory feedback.

Conclusions From Research on Cybernetic Anthropomorphic Machines

One of the first comprehensive conceptual and technical investigations of human-factors engineering requirements for effective workstation telepresence, as defined generally in Figures 1 and 2, was carried out during the fifties and sixties in the context of an extensive General Electric research and development program on cybernetic anthropomorphic (manlike) machines (CAMs) [11].

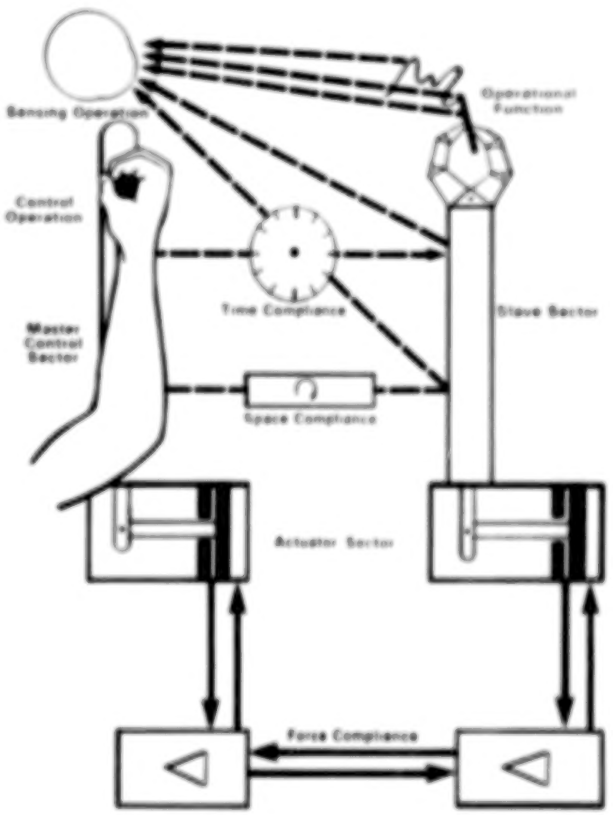


Figure 1. Human factors in control-display relationships in a manual manipulator. Normal spatial and temporal visual feedback compliance is essential for optimal performance.

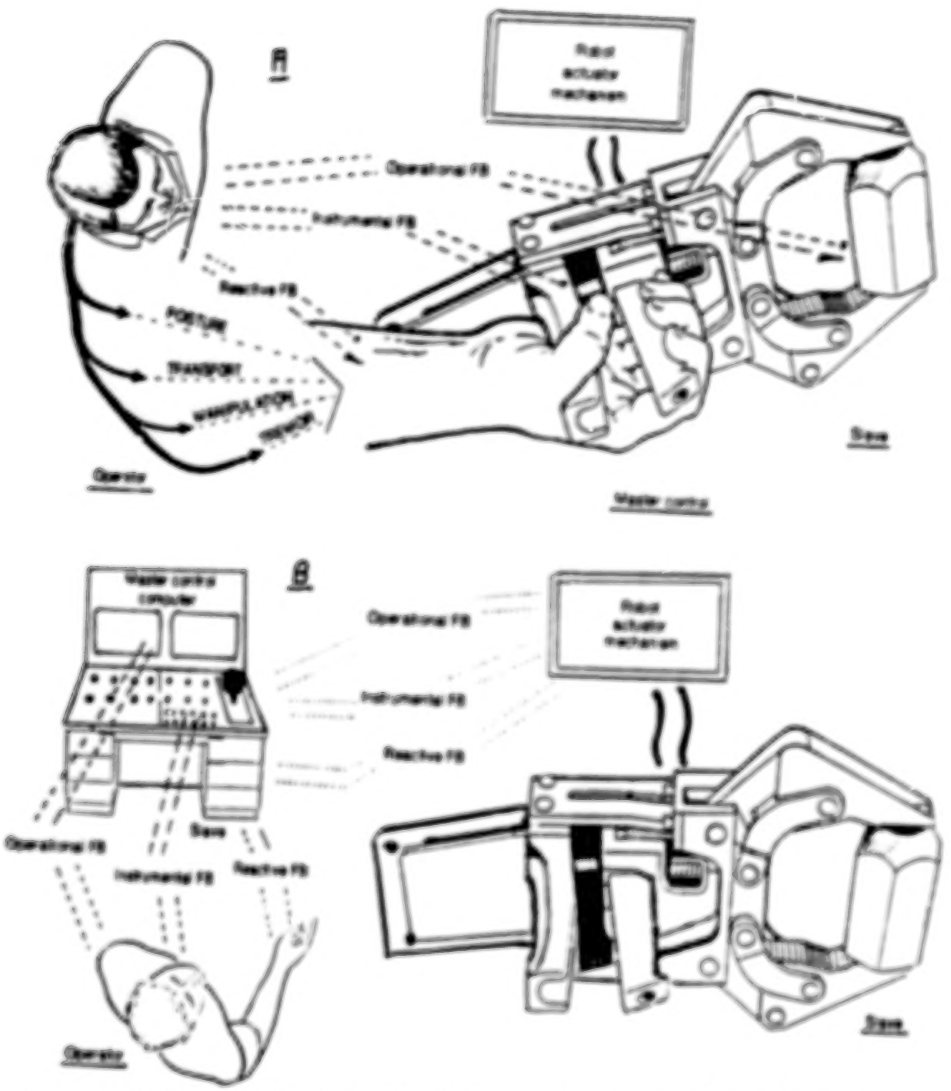


Figure 2. Sources of visual feedback in direct versus remote operation of a manual manipulator. Reactive feedback is from movement of limb or segment effector. Instrumental feedback is from effector-mediated movement of master-control component. Operational feedback is from action of slave component. A. Direct operator interaction with telerobot; B. Computer-mediated operator interaction with telerobot.

under the direction of Mosher [17,18]. The broad objective of the program was to develop mechanical devices which would serve as operator-controlled extensions of the body to expand the strength and endurance capabilities of human performance. The program dealt with four general types of CAMs: arm-claw manipulators, bipedal walking machines, quadripedal walking machines, and ambulatory exoskeletons which enclosed the operator within body-like frames. These were true telerobotic devices, in that they each comprised master control, actuator, and slave subsystems which provided computer-mediated, servo-controlled spatial, temporal, and force feedback compliant with all modes and patterns of motorsensory feedback from the human operator's articulated arm, hand-wrist, leg, foot, and torso movements.

To measure system relationships between operator performance and machine feedback parameters (such as device position, servo gain, and force-feedback ratio), human-factors studies of the four types of CAMs were carried out. These studies were guided by the general block diagram of an operator-computer-telerobot system illustrated in Figure 3. Figure 3 is based on the original design for a CAM system [18]. However, the figure has been adapted [11] to show a more detailed breakdown of the operator-workstation-telerobot interactions schematically illustrated in Figure 2B, including specifically the sources of reactive, instrumental, and operational sensory feedback provided to the operator from the master-control, actuator, and slave subsystems. The figure also has been modified to conform to current specifications [31,32,64] for the Space Station Flight Telerobotic Servicer (SSFTS), to be discussed further in a subsequent section.

Figure 3 emphasizes the complexity of sensory feedback relationships which must be built into telerobotic systems to provide fully compliant force and position feedback for effective telepresence. Force feedback to the operator is mediated by kinesthetic (also termed proprioceptive) mechanoreceptors in the muscles, tendons, and

joints, plus tactile receptors in the skin. Collectively, kinesthetic and tactile receptors provide muscle-tendon-joint sensibility, which encompasses pressure, movement position and dynamics (occurrence, locus, direction, velocity, and acceleration of movement), temperature, and pain [19, Chap. 14].

Mosher [17,18] established that four different dimensions of force feedback are needed to control body movement accurately, and should therefore be incorporated into a telerobot control system: (1) force of movement at different levels of exertion; (2) upper limit of force required; (3) prediction of force needed to execute movements of a given velocity and power; and (4) detection of relative displacement between movement and its sensory feedback. A series of system design and operational factors which influence the muscle-tendon-joint sensibility of the operator were identified, including force feedback ratios between operator and robot, drift on bias forces, friction thresholds, nonlinearity of force interactions, saturation factors, and force-signal integrity as influenced by system kinematics. As part of this human-factors system analysis, Mosher also demonstrated that force feedback related to the limits of exertion of powered grips, lifts, and pulls must be built into telerobots as a safety precaution to prevent their destructive action, a finding with implications for present-day robotic systems [20].

Although theoretical and experimental demonstration of the fact that force feedback is needed to guide and control telerobotic machines was fully developed and confirmed by the work of Mosher (17,18), research conducted by the senior author of the present report established that effective workstation telepresence could not be achieved without also introducing spatial compliance in sensory feedback control of telerobotic action [11,21,22]. Such compliance occurs only when the position feedback system designated in Figure 3 embodies several spatiotemporal parameters of sensory feedback, namely position, direction, relative extent, range, velocity, acceleration, and form or pattern of telerobotic movement.

As described in the next section, the roles of spatiotemporal sensory feedback factors in human movement control were established in the course of an extensive program of experimental research on the effects of spatially and temporally perturbed sensory feedback on human behavior and performance [14, 23-29]. Findings from this research

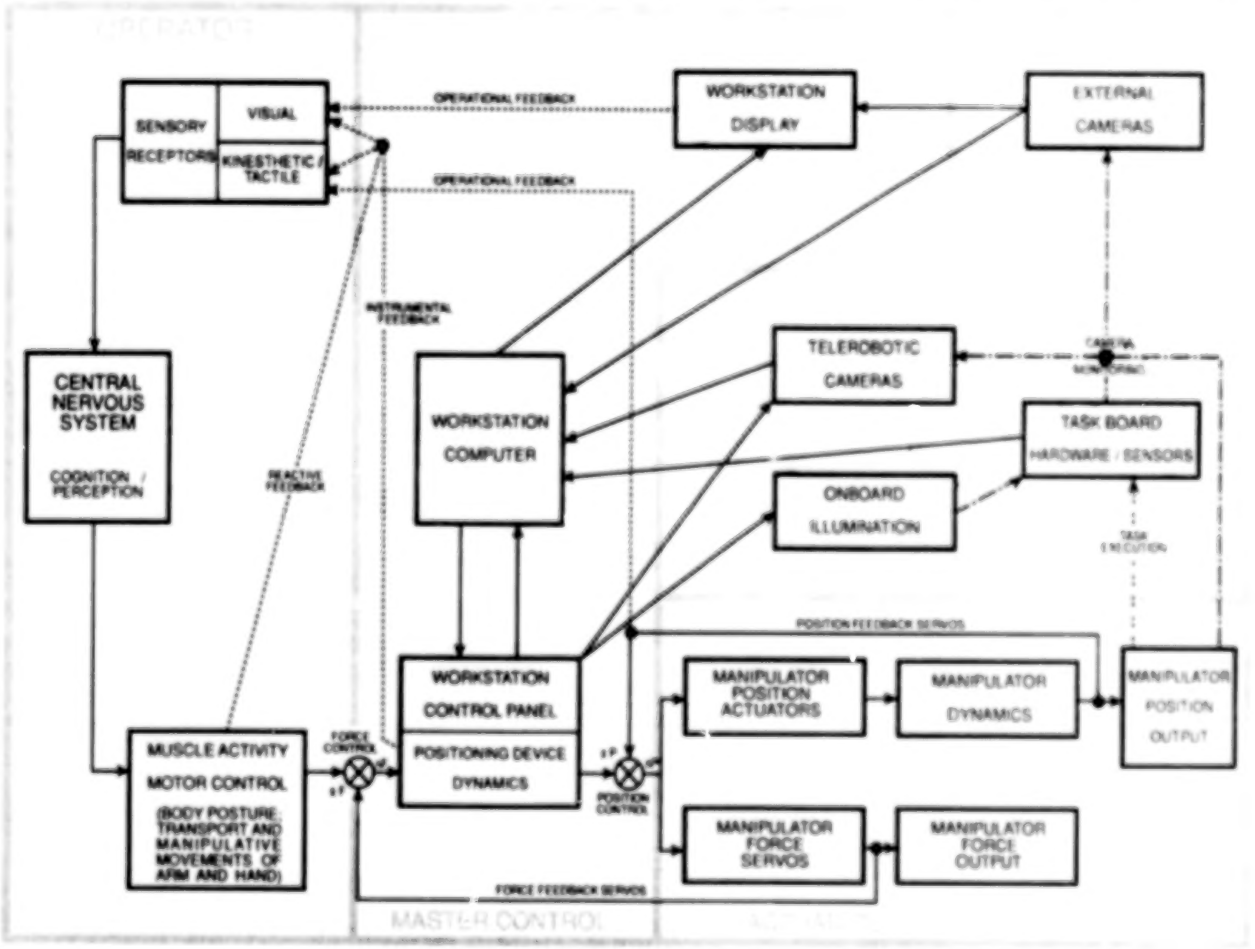


Figure 3. Block diagram of telerobot control system, indicating interactive roles of the operator, workstation, telerobot, and video cameras in system performance. Visual, position, and force feedback to the operator combines reactive, instrumental, and operational feedback generated by the master-control, actuator, and slave components of the system (Figure 2). The figure has been modified (from [11,18]) to include vision system components for a space telerobot [32].

demonstrate that all body movements are guided and controlled primarily by spatial feedback factors, with different movement systems mediated by different properties and dimensions of spatial feedback control. In general, tactile-kinesthetic mechanisms (for velocity, acceleration, and position control) are integrated with the visual and vestibular mechanisms (for control of position, direction, range, extent, and pattern of movement in space) in mediating the overall organization, predictive guidance, and integration of every movement pattern of the body.

Several characteristics of Mosher's original CAMs point up the importance of full spatial compliance for effective telepresence. For example, one problem that emerged in the design of the bipedal walking machines was inability of the operator to determine whether the machine was going down an incline versus falling backwards (or going up an incline versus falling forwards). To resolve this problem, Smith [21] demonstrated, first, that the operator's normal posture had to be maintained with respect to leg motion, and second, that the operator had to be able to sense any displacement of either leg of the machine with respect to the position of the machine cab. It was concluded that these cab-leg displacements had to be sensed by the operator via active force and position feedback. After appropriate torso-leg/cab-leg servo feedback mechanisms were incorporated into the design of the walking machine, it was found that even a novice operator could effectively control the dynamic balance of the machine.

A second example of the importance of spatial factors in telepresence emerged with research on manipulator CAMs. Tests showed that the operator of a manipulator experienced extension of his/her dynamic body image as long as just the manipulator was moved, but that this sensation was lost when the manipulator claw was moved. This loss of telepresence was related to the fact that the grasp actions of the claw were not spatially compliant with the operator's hand movements, although the force compliances were very adequate.

The two examples cited above demonstrate that body image is defined in large part by the spatial organization of body movement, and that the skill, precision, and safety of telerobotic operation are enhanced when the human operator perceives robot movements as an extension of his/her own movements. The general conclusion is that telerobotic design must incorporate compliances of both spatial and force factors between operator movements and machine movements before effective workstation telepresence can be established.

Although this conclusion is based on CAM research dating back two to three decades [17,18,21,22], we are not aware of more recent evidence which contradicts the insights it provides into the human factors of workstation telepresence, or that superior approaches to telerobotic design and control have been devised. Indeed, a recognized expert in the field recently referred to Mosher's machines as still among the most sophisticated telerobots yet developed [30].

EFFECTS OF PERTURBED SENSORY FEEDBACK ON INTERACTIVE PERFORMANCE

The remainder of this report summarizes scientific evidence from perturbed sensory feedback research which indicates why spatially compliant feedback is critical to workstation telepresence, and explores the implications of this evidence for optimizing the human-factors design of interactive, telerobotic systems. Scientific interest in the phenomenon of perturbed sensory feedback dates back over a century to the early work of Helmholtz [33], who studied the effects of and adjustment to spatial displacements in visual feedback. Since then, the effects on human performance of both spatial perturbations and temporal delays in visual and auditory feedback have been the subject of numerous investigations, whose findings are summarized in a series of reviews published over the past three decades [14, 23-29,34].

In this report, the conceptual approach adopted to interpreting the performance effects of perturbations in sensory feedback is grounded in behavioral cybernetic theory. The basic postulate of this theory is that behavior is a closed-loop, self-regulated process mediated by motor control of sensory feedback [12,14,23-29]. From a behavioral cybernetic perspective, sensation and perception are feedback integrated with the motor system through motor coordinate feedback control of receptor processes and environmental stimuli. During task performance, the individual uses movement-based mechanisms to control spatial, temporal, or other types of perturbations in sensory feedback which may arise as a result of the human-factors characteristics of task design.

Since Helmholtz first posed the question, an important issue in psychology has been whether the spatial organization of behavior is innate or learned through experience. Although experiential learning appears to definitely contribute to the spatial guidance of behavior, experimental findings indicate that complete adaptation to perturbations in sensory feedback rarely occurs [14,23-29,34]. Instead, evidence indicates that performance and learning remain suboptimal in the face of persistent exposure to a perturbation condition. This suggests that the nature and degree of learning and performance proficiency achieved for a particular task depend upon the nature and degree of spatiotemporal compliance between sensory input and the motor feedback mechanisms controlling that input during task execution.

Figure 4 provides a diagrammatic illustration of the concept that workstation telepresence is critically dependent upon spatial compliance between sensory feedback and its motor control mechanisms, using the same master control-slave telerobot model as in Figure 1. In Figure 4, however, visual feedback has been spatially perturbed by reversing the lateral motion relationships between arm-controlled movement of the control component and derivative movement of the slave component. Consequently, the operator experiences a perturbation in the normal manner in which visual feedback from the telerobot manipulator is controlled by arm movement, telepresence is compromised, and performance suffers.

In general, four major classes of perturbations in sensory feedback may be identified, namely: (1) spatial transformations and/or displacements; (2) temporal delays; (3) changes in energy properties; and (4) modality conflicts. For example, control of visual feedback from a video-

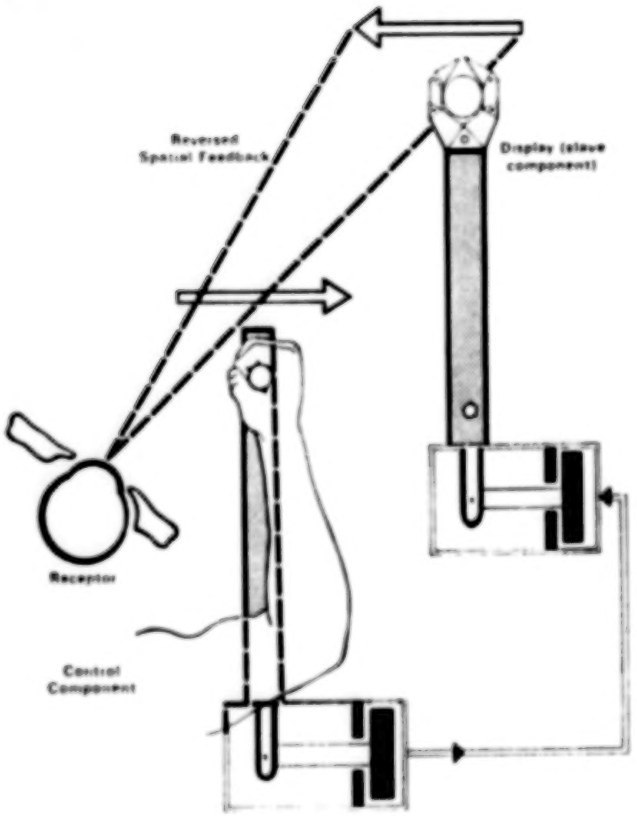


Figure 4. Spatial perturbation (reversed visual feedback) in visual-manual operation of a telerobot manipulator. Spatial compliance between movements of telerobot slave components, and control movements of the operator, represents one of the key human factors of workstation telepresence.

displayed image during remote telerobotic operation may be perturbed in each of these ways by: (1) inversion, reversal, angular displacement, magnification, or miniaturization of the displayed image; (2) feedback delay of the real-time display; (3) modulation of display quality due to glare, brightness, reflectance, contrast, and/or frequency shift effects; or (4) auditory and/or kinesthetic interference with visual feedback control. In extraterrestrial environments, microgravity represents a fifth class of sensory feedback perturbation which may exacerbate the effects of other types of perturbations [2,35-37].

Figure 5 expands upon Figure 4 with a schematic summary of the major sets of factors which potentially influence visual-manual task performance and workstation telepresence during operator interaction with a space telerobot such as the SSFTS. The diagram indicates that interaction of the operator (to the right) with the telerobotic system (to the left) across the operator-system interface is influenced by one or more of 14 possible perturbed sensory feedback factors specified in the figure and cited above. The operator-system interface specified in Figure 5 assumes the feedback relationships depicted in Figures 2 and 3, namely 3 sources of feedback (reactive, instrumental, operational) from the 3 telerobotic subsystems (master-control, actuator, slave).

Figure 5 also assumes that visual-manual performance is based upon use of the motor system (19 principle motor control factors specified) to feedback control the visual system (11 principle visual perception factors specified), involving 5 distinct modes of sensory feedback control [14,23-29]. This control process potentially may be influenced by a broad range of space telerobotic system design factors (18 principle design considerations specified), some of which may give rise to the perturbation conditions indicated in the figure. The system design, visual perception, and movement control factors specified in Figure 5 are compiled from basic human-factors considerations regarding telerobotic operation [11], plus recent analyses by various scientific specialists in space telerobotic research [32,38-41].

Some appreciation of the scientific challenge confronting systematic investigation of the human factors of workstation telepresence may be gained by considering that there are over 2.3 million possible combinations of performance, design, and feedback factors (14x3x3x19x11x5x18) which potentially can influence operational variability in visual-manual control of a space telerobot, based on specifications in Figure 5 and summarized above. That it probably is not feasible to completely evaluate even a reasonable subset of such possible combinations raises the question of research priorities and emphasis in human-factors analysis of performance variability in telerobotic operation. The hypothesis adopted in this report is that spatiotemporal perturbations in sensory feedback constitute a major source of variability in human performance with automated and telerobotic systems, and that this issue therefore merits attention by human-factors research aimed at optimizing workstation telepresence. Evidence in support of this hypothesis is summarized in the next section.

Experimental Observations On Perturbed Sensory Feedback Effects

Over the past three decades, the most extensive program of experimental research on perturbed sensory feedback phenomena has been conducted by K.U. Smith and colleagues [14,23-29,34]. This work has focused primarily upon delineating the effects of spatial and temporal perturbations in sensory feedback on learning and performance. For this purpose, computer- and video-based techniques are used to introduce controlled spatial transformations or displacements, and/or temporal delays, in sensory feedback (usually visual or auditory) provided to subjects of their own task performance. A wide range of different cognitive and motor behavioral skills have been examined in this manner, including machine performance, tool using, musical performance, reading, writing and drawing, tracking and steering, speech, memory, eyemovement control, visual perception, motor coordination, postural control, behavioral-physiological integration, and social interaction [28,29,42,43].

With its use of computer and video-display techniques, this research essentially represents the application of interactive, human-computer laboratory methodology to the study of how perturbations and distortions in sensory feedback

influence learning, performance, and workstation telepresence. However, rather than arising as an incidental and unwelcome concomitant of human-factors defects in interface design, such perturbations are introduced deliberately by the computer under controlled laboratory conditions, so that effects can be assessed in an objective and quantitative fashion. The research approach therefore provides a methodological paradigm for systematic study of the contribution of sensory feedback perturbations to variability in human interaction with automated and telerobotic systems, in relation to human-factors design features of the interface [11,14].

The general conclusion from the extensive body of research conducted using this approach is that the performance and learning of every behavioral task so far examined is degraded by perturbations in sensory feedback [14,28,29]. Tables 1 and 2 provide representative data on the nature and extent of these effects, for spatial transformations and temporal delays in sensory feedback respectively. All data are derived from controlled laboratory study of subject interaction with computer-or video-based systems. In both tables, performance changes for a series of tasks under specified perturbation conditions are summarized, along with information regarding experimental conditions and reference citations (first column).

Under spatially perturbed visual feedback conditions, results for various measures of seven distinct types of tasks are given in Table 1 (third column), namely writing, drawing, reading, assembly, panel control, tool using, and tracking tasks. The spatial perturbation conditions examined (second column) encompass inversion, reversal, combined inversion and reversal, angular displacement, and size reduction (miniaturization) of visual feedback. The measures used to assess performance (fourth column) vary for different tasks, but generally fall into the categories of movement characteristics (contact, travel, grasp, or assembly times), or performance accuracy. Results (last column) are expressed as percent change in performance, from the control to the spatial perturbation condition, at the start and at the end of a training period (fifth column) which typically involved one or more trial sessions per day over a series of days. An asterick is used in the last column to denote the statistical significance of the performance change observed.

Results for the feedback delay studies in Table 2 are summarized in much the same manner. These studies examined writing, drawing, tracking (manual, eye, and head movements), speech, memory, and interactive social tracking tasks (second column), with performance again assessed using either movement characteristics or accuracy measures (third column). Results (last three columns) are presented as the change in performance, for delay in either visual or auditory feedback, at delay intervals of 0.2 seconds, 0.4 seconds, and the

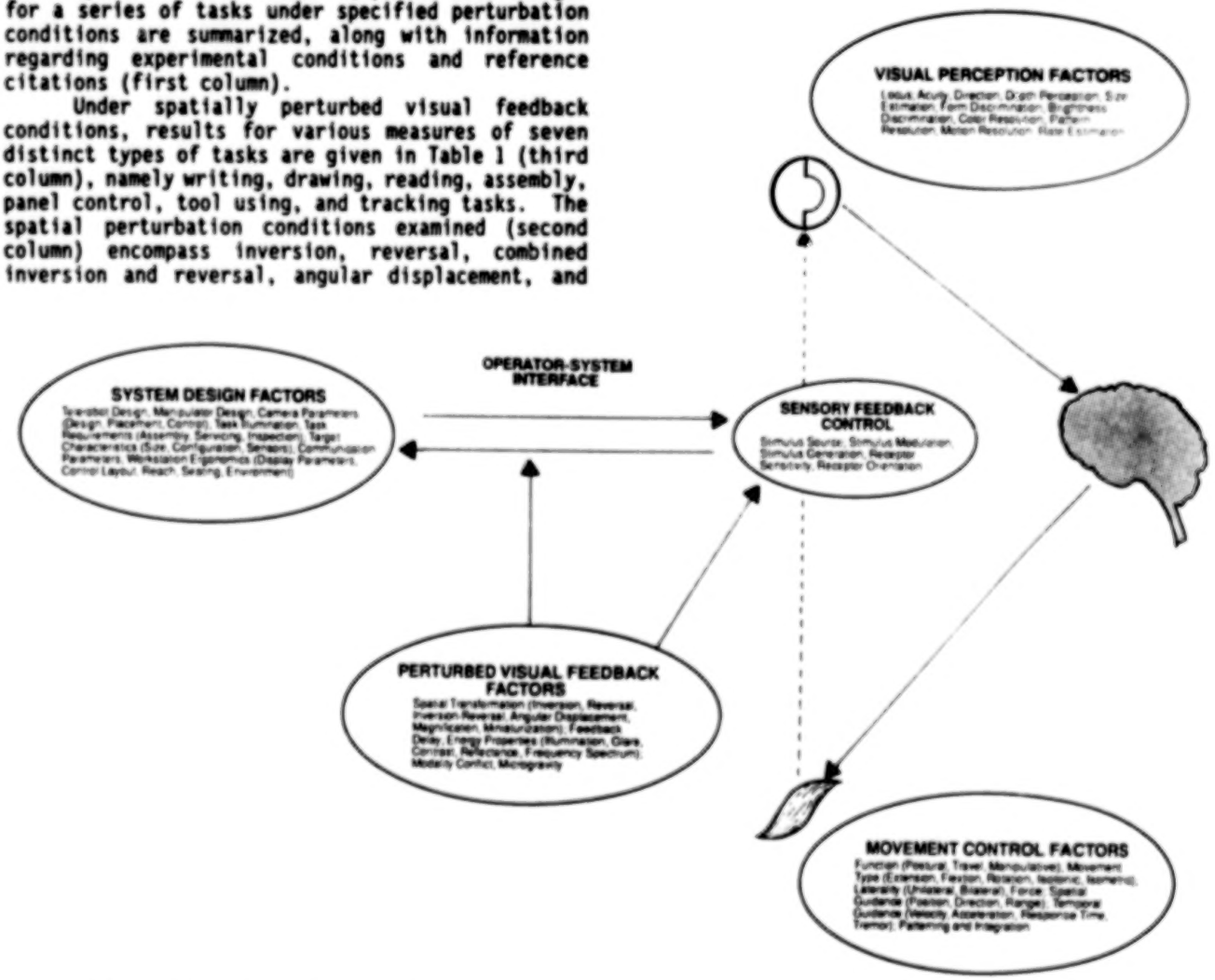


Figure 5. Human factors which may influence workstation telepresence in space telerobotic operation. Operator performance involves motor control (movement and sensory feedback control factors) of visual feedback (visual perception factors), which is influenced by both system design and perturbed visual feedback factors.

maximum delay interval examined in the study, relative to performance under the no delay condition. Performance change values for the delay conditions are expressed as multiples of the zero-delay levels. In all of the studies cited in Table 2, analysis of variance was used to assess the overall contribution of the delay condition to the performance variability observed. In every case, the overall contribution was found to be statistically significant, but tests of significance at individual delay levels were not performed.

Results summarized in Tables 1 and 2, plus findings from an extensive body of related research, support the following conclusions regarding the effects of spatial and temporal perturbations in sensory feedback on interactive performance [14,23-29].

1. Performance and learning of every behavioral task so far examined are degraded by such perturbations. Consistent patterns and features of behavioral disturbance appear from study to study. Oscillatory instability in movement control becomes more pronounced, accompanied by increased variability and extremes in movement velocities and accelerations. Accuracy of movement guidance and tracking declines. Perception is disturbed and, in some cases, may disappear entirely; there is a concomitant deterioration in learning. Subjects report feeling confused, uncertain, and/or uncomfortable about their behavior. Skilled performers are particularly sensitive to these effects.

Table 1. Changes in Interactive Performance and Learning Under Spatially Perturbed Visual Feedback

Reference-Subjects [1]	Perturbation Condition [2]	Task	Measure	Days of Training (Trials) [3]	Performance Change (%) [4]	
[44] - 12	I	.Hand-	.Contact	0 (1)	+ 676 %	*
"	"	writing	Time	20 (20)	+ 153	*
"	"	"	.Travel	0 (1)	+ 410	*
"	"	"	Time	20 (20)	+ 8	
"	"	.Drawing	.Contact	0 (1)	+ 493	*
"	"	Triangles	Time	20 (20)	+ 71	*
"	"	"	.Travel	0 (1)	+ 583	*
"	"	"	Time	20 (20)	+ 33	
"	"	.Reading	.Amount	0 (1)	- 81	*
"	"	Words	Read	5 (5)	- 12	*
[45] - 4	I	.Reading	.Time x	0 (1)	+ 253	*
"	"	"	Error	5 (5)	+ 21	
"	R	"	Score	0 (1)	+ 210	*
"	"	"	"	5 (5)	+ 6	
[24] - 10	I-R	.Assembly	.Grasp	0 (1)	+ 5	*
(p. 163)	"	"	Duration	4 (4)	+ 4	*
"	"	"	.Loaded Tra-	0 (1)	+ 24	*
"	"	"	vel Duration	4 (4)	+ 6	
"	"	"	.Assembly	0 (1)	+ 92	*
"	"	"	Duration	4 (4)	+ 60	*
"	"	"	.Nonloaded	0 (1)	+ 27	*
"	"	"	Travel Duration	4 (4)	+ 8	
[24] - 24	I	.Hand-	.Contact	2 (4)	+ 286	*
(p. 171)	R	writing	Time	"	+ 154	*
"	I-R	"	"	"	+ 250	*
"	I	"	.Travel	"	+ 240	*
"	R	"	Time	"	+ 160	*
"	I-R	"	"	"	+ 200	*
"	I	.Drawing	.Contact	"	+ 171	*
"	R	Triangles	Time	"	+ 80	*
"	I-R	"	"	"	+ 105	*
"	I	"	.Travel	"	+ 139	*
"	R	"	Time	"	+ 72	*
"	I-R	"	"	"	+ 94	*
[24] - 24	I	.Maze	.Accuracy	0 (1)	- 72	*
(p. 172)	R	Tracing	"	"	- 51	*
"	I-R	"	"	"	- 39	*
"	I	"	"	3 (9)	- 37	*
"	R	"	"	"	- 15	*
"	I-R	"	"	"	- 25	*

Table 1. (continued)

Reference-Subjects [1]	Perturbation Condition [2]	Task	Measure	Days of Training (Trials) [3]	Performance Change (%) [4]	
[24] - 18	.Size	.Panel	.Contact	0 (1)	+ 108	*
(p. 218)	Reduction	Control	Time	5 (40)	+ 104	*
"	(to 1/3 size)	"	.Travel	0 (1)	+ 232	*
"	"	"	Time	5 (40)	+ 267	*
[24] - 24	I	Tool Using:				
(p. 239)	"	.Dial	.Manipulation	0 (1)	+ 273	*
"	"	Setting	Time	7 (21)	+ 117	*
"	"	.Turning	"	0 (1)	+ 191	*
"	"	"	"	7 (21)	+ 78	*
"	"	.Pushing	"	0 (1)	+ 200	*
"	"	"	"	7 (21)	+ 100	*
"	"	.Pressing	"	0 (1)	+ 250	*
"	"	"	"	7 (21)	+ 117	*
"	"	.Dial	.Travel	0 (1)	+ 257	*
"	"	Setting	Time	7 (21)	+ 29	*
"	"	.Turning	"	0 (1)	+ 209	*
"	"	"	"	7 (21)	+ 40	*
"	"	.Pushing	"	0 (1)	+ 218	*
"	"	"	"	7 (21)	+ 36	*
"	"	.Pressing	"	0 (1)	+ 230	*
"	"	"	"	7 (21)	+ 40	*
[46] - 24	AD:					
"	20 degrees	.Maze	.Errors	0 (1)	+ 6	
"	40 degrees	Tracing	"	"	+ 5	
"	60 degrees	"	"	"	+ 23	*
"	10 degrees	"	"	10 (10)	- 2	
"	20 degrees	"	"	"	- 9	
"	30 degrees	"	"	"	+ 2	
"	40 degrees	"	"	"	+ 7	
"	50 degrees	"	"	"	+ 63	*
"	60 degrees	"	"	"	+ 148	*
"	70 degrees	"	"	"	+ 287	*

*Change from nonperturbed visual feedback performance level is statistically significant ($p<.05$).
[1]Reference in brackets, followed by number of subjects.
[2]I = Inversion; R = Reversal; I-R = Inversion-Reversal; AD = Angular Displacement (in degrees).
[3]Number of trials in parentheses.
[4]Performance change, in percent, calculated as:
[(perturbed viewing level - nonperturbed viewing level) x 100] / (nonperturbed viewing level)

2. For untrained subjects, spatial perturbations in visual feedback have profound effects on task performance. In Table 1 for example, there are statistically significant performance decrements for 24 of 27 no training cases (fifth column, trial 1, 0 days training), and in 20 of these cases the performance decrement exceeds 50 percent.

3. After training, subjects typically exhibit improved performance under spatially perturbed visual feedback conditions. In Table 1 for example, there are 27 instances in which pre- and post-training performance levels can be compared, and performance improved in 25 of these cases. Nevertheless, with rare exceptions, under spatially perturbed visual feedback subjects do not perform at the same level as under nonperturbed conditions, even after a training regimen lasting as long as 20 days. In Table 1 for example, statistically significant performance decrements persisted for 33 of 43 cases in which training was provided for 2 days or longer. A similar pattern has been observed in experiments where subjects were exposed continuously to spatially perturbed visual feedback for periods of time lasting days or weeks [24,34].

4. The training findings summarized in Point 3 suggest that human learning is refractory to complete adaptation to spatial perturbations in visual feedback. Neurobiological corroboration of this conclusion is provided by recent research [52,53] on motor learning in the primate vestibulo-ocular reflex (VOR). Under continuous exposure to size distortion (magnifying or miniaturizing spectacles), both the reflex gain and activity of brain stem neurons mediating the reflex adjust accordingly. However, the gain error increases from about zero, with no distortion, to 10-20 percent under the distortion condition. This lack of complete adaptation to size perturbation may explain

Table 2. Changes in Interactive Performance Under Feedback Delay[1]

Reference-Subjects[2]	Task	Measure	Performance Change (xl) at Specified Delay[3] 0.2 sec	0.4 sec	Max (sec)[4]
[23] - 2	.Writing Letters	.Contact Time	-	-	+ 2.4x (.52)
"	.Drawing	"	-	-	+ 1.1x (.52)
"	.Star Tracing	"	-	-	+ 4.6x (.52)
"	.Maze Tracing	"	-	-	+ 5.0x (.52)
[47] - 8	.Head Movement Tracking of Visual Target	.Tracking Error	+ 1.3x	+ 1.6x	+ 2.3x (0.8)
[48] - 8	[auditory delay interval[5] :		|0.1 sec|	0.2 sec|]	
	.Speech	.Speech Errors	|+ 4.1x |	+11.1x |	+14 x (0.3)
[49] - 12	Tracking of Visual Target:				
	.Eye Movement	.Tracking	+ 1.2x	+ 1.2x	+ 2.0x (1.6)
	.Head Movement	Error	+ 1.2x	+ 1.2x	+ 5.0x (1.6)
	.Head-Eye Movement	"	0 x	+ 1.2x	+ 1.9x (1.6)
[50] - 8	.Memory of Visual Image	.Recall Error	+ 1.5x	+ 1.9x	+ 3.3x (0.8)
[51] - 2	.Visual-Manual Tracking	.Tracking Error			
	-Individual:	"	+ 1.1x	+ 1.5x	+ 2.6x (1.5)
	-Social :	"	+ 1.5x	+ 1.8x	+ 3.0x (1.5)

[1]All studies examined visual feedback delays, except for Abbs and Smith [48], who examined auditory feedback delay.
[2]Reference in brackets, followed by number of subjects.
[3]Performance change, in multiples of one (indicated by small x), observed at delay intervals of 0.2 sec, 0.4 sec, and maximum delay interval examined[4]. Performance change is calculated as:

(Delayed Performance Level) / (No Delay Performance Level)

[4]Performance change, in multiples of one[3], at maximum feedback delay interval examined in study. The maximum delay interval examined, in seconds, is indicated in parentheses.
[5]Delay intervals of 0.1 sec, 0.2 sec, and 0.3 sec examined in this study.

the finding cited in Table 1 that performance by human subjects in an interactive panel control task remained significantly impaired after 5 days of training with use of miniaturizing spectacles that provided a two-thirds size reduction [24, p. 218].

5. Three experiments described in Table 1 [24, pp. 171-172; 45] directly compared the effects of inversion, reversal, and combined inversion-reversal of visual feedback. The results indicate that inversion has the most adverse impact on performance, followed in order by combined inversion-reversal, and reversal alone. Other research also supports this conclusion [25].

6. Under conditions of angular displacement of visual feedback, as indicated in the last study cited in Table 1 [46], performance remains relatively unaffected until a displacement angle of 50 degrees is imposed. This finding has been interpreted as providing evidence for a breakdown angle of spatial displacement, beyond which performance becomes progressively more difficult. Typically, the breakdown angle falls in the range of 40 to 50 degrees displacement. Results in Table 1 indicate that even after 10 days of training, performance under displacement conditions exceeding the breakdown angle remains significantly impaired.

7. Relative to direct viewing conditions, performance on an assembly task is impaired under television viewing of the assembly operation, even when no deliberate spatial distortions are introduced [24, pp. 161-163]. This may be attributed to inherent spatial distortions in visual feedback introduced by the video camera and display system.

8. Performance under spatially perturbed visual feedback varies as a function of both age and gender [45,54].

9. Temporal perturbations (feedback delays) in sensory feedback also result in consistent decrements in performance, as illustrated by the findings cited in Table 2. Performance decrements at a delay interval of 0.2 sec occur for eight of nine tasks listed in the table. At delay intervals of 0.3 sec and greater, all 13 tasks listed in the table manifest performance decrements. For speech, a fourfold increase in errors is observed at an auditory feedback delay of 0.1 sec [48].

10. Training effect data are not included in Table 2 because the studies that have been done (on both eyemovement tracking and memory) indicate little or no performance improvement with learning under feedback delay conditions [47,55].

11. When visual feedback is both temporally delayed and spatially reversed in either an eye- or a head-movement tracking task, the delay exacerbates the decrement in tracking performance produced by the reversal condition [47,55].

12. Social tracking involves the mutual exchange and control of sensory feedback among two or more individuals during group activity [13,28,56]. Salient findings regarding social tracking are that: (a) relative to visual-manual social tracking, accuracy is higher for tactile or auditory social tracking [57]; (b) social tracking accuracy is comparable to individual tracking accuracy except under feedback delay conditions, when the latter is superior to the former at all delay levels [51]; (c) learning of social tracking tasks is limited, highly variable and inconsistent, and unstable [58,59]; and (d) the preponderance of performance variability in social tracking is attributable, not to a learning effect, but to human-factors characteristics of task design [60].

Space Telerobotic Telepresence Operational Implications of Perturbations in Sensory Feedback

A major assumption of this report is that the findings and conclusions summarized in the preceding section have direct relevance to understanding possible effects of perturbed sensory feedback on interactive performance in extraterrestrial environments. For example, many of the tasks listed in Tables 1 and 2 (reading, tracking, assembly, panel control, tool using, speech, memory, social tracking) are employed during interactive human performance with space telerobotic systems. Smith and colleagues [61] have addressed this question, and conclude that perturbed sensory feedback effects could have a potentially decisive influence on the performance of useful work with man-in-the-loop space telerobotics. Some experimental evidence from earth-based, human-factors research on telerobotic systems lends support to this conclusion, as summarized below.

Establishment of high fidelity workstation telepresence for space telerobotic operators will require that inappropriate or hazardous sensory feedback from the space environment be eliminated (i.e., direct sunlight, glare, cold, lack of air), whereas visual, tactile, kinesthetic, and auditory feedback essential to manipulative performance is provided which conforms to the control capabilities of the operator. During performance of a remote manipulation task with a telerobot in space, it may be possible to view the remote scene directly through a window. For most tasks, however, a direct view either will not be available, or will not provide the appropriate visual cues necessary for teleoperation (due to distance and/or interposition of other objects). For this reason, external cameras will provide the primary source of visual feedback about manipulator position, orientation, and rate of movement [62].

Various studies of viewing system requirements for space telerobotic systems indicate that perturbations in visual feedback may represent a pervasive feature of teleoperation. A number of authors agree [38-41] that visual guidance of space station telerobotic tasks will be influenced by extreme illumination, contrast, glare, and reflectance conditions known to exist in space. In addition, recent research by R. Smith and colleagues [32,39,62,63] has addressed the question of how spatial perturbations in visual feedback, produced by camera-induced transformation of the manipulator reference plane relative to that of the operator, could degrade task performance of the telerobot operator and thereby risk damage or loss of the manipulator or payload. In particular, spatial perturbations of visual feedback to the operator from a space telerobot can be predicted as a result of: (1) variable orientations of surveillance cameras on the station gridwork and modules, and on the telerobot; (2) lack of a fixed ground reference; (3) abnormal or missing depth cues; and (4) extreme viewing conditions, as noted above.

Decremental effects of spatial transformations in visual feedback on remote operation of a telerobot have been confirmed in a recent laboratory study by Stuart and Smith [63], who show that telerobot manipulation times are: (1) significantly slower for video-display viewing of the worksite, relative to direct viewing; and (2) slowed most by inversion of visual feedback, with reversal and inversion-reversal conditions having less of an effect. In particular, these researchers found that performance completion times in a manipulator positioning task were increased by the following amounts, relative to direct viewing of the task: (1) display viewing, 2.0-fold increase (no other spatial perturbation); (2) inverted-reversed viewing, 8.5-fold increase; (3) reversed viewing, 10.2-fold increase; and (4) inverted viewing, 16.1-fold increase. As indicated in Table 1, these findings recapitulate the earlier observations of K.U. Smith and colleagues [24], in which the same techniques of video camera displacement and rotation were used to spatially perturb visual feedback of the task provided to subjects.

Also recapitulating earlier observations [23] are results from recent research on operation of a laboratory version of a telerobot servicer [15], which demonstrate that delays in visual feedback significantly increase manipulation performance times. The earlier research documented the detrimental effects on tracking performance of delay intervals ranging from one to three seconds (in the same range as the round-trip transmission delay for a signal from an earth-bound operator to an orbiting or moon-based telerobot), and was among the first to point out that effective guidance of a remote telerobot would be seriously compromised by delays in visual feedback [23, pp. 94-96].

A further important consideration in establishing telerobot workstation telepresence is that of sensory feedback modality and its control. For example, in an experimental study of visual and tactile social tracking [25], it was shown that an observer can track an irregularly moving object more accurately by touching it (tactile-kinesthetic feedback) than by watching it (visual feedback). The difference is attributable to the slower feedback relationships of the visual system as

compared with those of the tactile-kinesthetic system. There also is the question of modality compatibility between sensory feedback and sensory feedback control. For example, feedback of force information from a telerobot force transducer to a hand controller, rather than as visual or auditory feedback, is compatible with human kinesthetic senses that are directly interpretable.

An appreciation of the potential for spatiotemporal perturbations in sensory feedback during SSFTS operation may be gained from a consideration of current system requirement specifications [31], as embodied in the general system diagram in Figure 3. Vision system specifications call for at least four telerobot-mounted video cameras, including one on the manipulator wrist, with separate positioning, orientation, zoom, focus, and aperture control capabilities indicated for each camera. Variable control of attitude and imaging parameters for both the telerobot-mounted and external cameras, coupled with possibilities for variability in task board and telerobot manipulator geometry, introduce a virtually unlimited potential for reversal, inversion, angular displacement, magnification, and/or miniaturization (Figure 5) of the display image presented as visual feedback to the human operator.

SSFTS manipulator specifications call for kinesthetic feedback of force, position, and rate information from sensors at each joint [31,64]. For purposes of task execution, the specifications also call for an orientation accuracy of less than 1.0 inch in manipulator tool plate center position, repeatable to an accuracy of less than 0.005 inch. However, findings summarized in Tables 1 and 2, plus observations from more recent perturbed sensory feedback research on telerobots [15,63], suggest that such accuracy may be compromised by the occurrence of any type of spatiotemporal perturbation in visual feedback. Under earth laboratory conditions, such perturbations cause errors in movement trajectory guidance plus increased oscillatory instability in static positioning control [23-29,64]. In addition, experience from earlier work on Mosher's CAM systems [11], summarized previously, suggests that it may be necessary to supplement kinesthetic and visual feedback with tactile feedback from manipulator end effectors in order to achieve a high degree of reproducible accuracy in control of manipulator positioning.

OPTIMIZING WORKSTATION TELEPRESENCE - A SOCIAL CYBERNETIC PERSPECTIVE

The analysis in the preceding sections suggests that effective workstation telepresence depends on provision of compliant sensory feedback across different modalities which can be readily controlled by the operator. Because they can compromise such compliance, perturbations in sensory feedback may significantly impair the operational effectiveness of space telerobotic systems, in relation to visual-manual control of manipulator functions. This raises the question as to what strategies might prove useful in reducing possible adverse effects of sensory feedback perturbations on human performance in space, in order to achieve high fidelity workstation telepresence. Training is one obvious choice. However, evidence cited in Tables 1 and 2 suggests that even extended training will not completely overcome these effects. It may therefore prove desirable, or even necessary, to develop techniques which enable the interactive computer to detect, and then correct, perturbations in sensory feedback before presentation of such feedback to the human operator.

The idea of enlisting the workstation computer as an intelligent co-participant in optimizing workstation telepresence is in line with proposals by various authors that the design of space telerobotic systems should evolve from primary reliance on operator control to reliance on both operator and autonomous, computer-mediated (robotic) control of task execution [1-3,65-67]. To provide a conceptual foundation for dealing with the human factors of workstation telepresence at various levels of interactive complexity, we have developed a social cybernetic model of human-computer interaction [12-14]. The model, shown in Figure 6A, assumes that behavioral cybernetic principles of human social interaction [28,56,58-60] also can be applied to an understanding of human-computer interaction.

The term social tracking used in Figure 6A refers to the feedback-controlled process by which an individual engaged in social behavior follows or tracks one or more social targets. During social tracking, the activities of one person in a social group effect behavioral-physiological changes in the other social partners, whose own activities in turn have behavioral-physiological feedback effects on the first. These effects arise as a consequence of control by each participant of sensory feedback generated by the other social tracking partners. During interpersonal social tracking for example, the movements of one individual generate sensory feedback that is controlled by tracking movements of the social partner, whose actions generate compliant sensory feedback for the first individual, who in turn tracks this social feedback with further movement, and so forth. The two social partners thus become dynamically yoked or interlocked behaviorally and physiologically, through mutual tracking and control of sensory feedback generated by each other's social behavior. Through such social tracking interlocks, participants in a social group begin to operate as an integrated system, with definite systems feedback parameters and control characteristics.

The premise of the model in Figure 6A is that the interactive computer can be considered as a machine analog of a human social partner, and can be imbued through adaptive programming with capabilities for tracking and controlling, across different modalities, sensory feedback generated both by its own activities as well as those of its human partner. Unfortunately, because of technological and design limitations in such capapabilities, today's interactive computer system is a relatively limited and impoverished social tracking target for the human operator. Broadly speaking, the creation of effective workstation telepresence requires that such limitations be overcome.

A number of authors have suggested that the quality and nature of crew social interaction are an important determinant of operational effectiveness

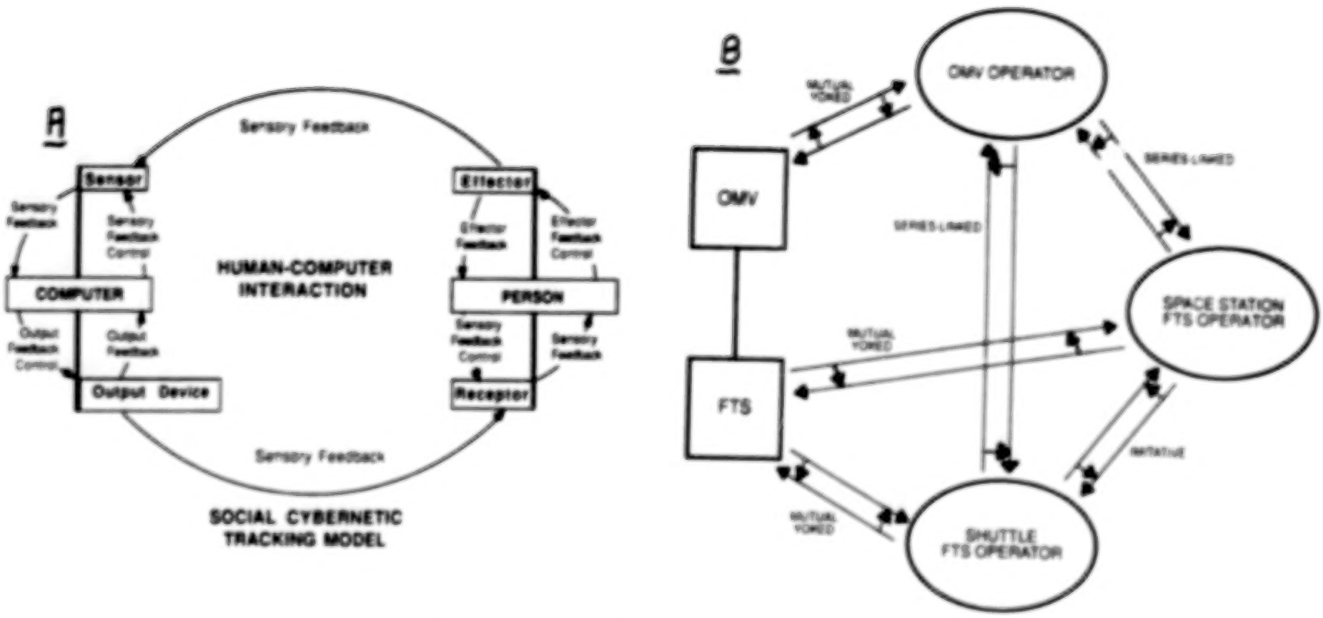


Figure 6. The social cybernetics of workstation telepresence. A. The social cybernetic model of human-computer interaction, which assumes mutual social tracking by operator and machine of sensory feedback generated by the other partner during interactive performance. B. Possible social cybernetic interactions during SSFTS operation. The diagram assumes that the SSFTS (which may be controlled by more than one operator) is conveyed to the task site by an Orbital Maneuvering Vehicle (OMV), controlled by a separate operator. The arrows symbolize possible sensory feedback exchange and control relationships between operator and machine participants in the control system. Also indicated are the most likely modes of social tracking for each relationship [13].

and safety in space [68]. Factoring possible human-computer interactions into the social tracking matrix further complicates the social cybernetics of flight crew performance. Figure 6B illustrates this point with a diagram of some of the interpersonal and human-computer social interactions which may be required for SSFTS operation. Current plans [1-3] call for the SSFTS to be conveyed to its task location by a transport device called the Orbital Maneuvering Vehicle (OMV). It is envisioned that the OMV and the SSFTS each will be controlled by a separate operator. In addition, current specifications [31] call for monitoring of telerobotic activities by multiple workstations (such as on the space station and shuttle), with the capability of handing control from one workstation to another. The various mutual social tracking interactions suggested by these plans are indicated in Figure 6B. The diagram also indicates the specific type of social tracking [13,56] which is most likely to be employed for the relationship specified.

The general implication of the diagrams in Figure 6 is that an analysis of the human factors of workstation telepresence requires consideration, not only of the social tracking fidelity for a particular operator-workstation-telerobot system, but also of the social tracking relationships between all of the players, human and computer, involved in system operation. That high fidelity social tracking requires effective control of sensory feedback from two sources---generated by self-activity and by activity of the social partner(s)---suggests two broad themes for human factors and engineering research aimed at improving workstation telepresence [12,13].

First, computer capabilities for detecting and adaptively controlling sensory feedback from its own sensors need to be enhanced. One useful outcome of such research could be automated, computer-mediated transformation of spatially perturbed images from cameras linked to the computer (see Figure 3). Secondly, computer capabilities for detecting and controlling sensory feedback from the human operator need to be enhanced. One useful outcome of such research could be improved techniques for automated detection of behavioral and physiological manifestations of operator decremental performance under spatiotemporally perturbed sensory feedback conditions, signalling need for computer-mediated correction of the perturbation. The human factors of workstation telepresence will benefit from the development of more robust strategies for automated control of sensory feedback, growing out of research on the social cybernetics of human-computer systems, which enable improved system management of spatiotemporal perturbations in sensory feedback.

SUMMARY

This report has presented a human-factors analysis of workstation telepresence which supports the following major conclusions:

1. Pioneering human-factors research on operational requirements for cybernetic anthropomorphic machines has established that spatial compliance between postural, transport, and manipulative movements of the operator and those of the CAM is essential to the creation of high fidelity workstation telepresence.

2. An extensive body of behavioral cybernetic research evidence indicates that spatial and temporal perturbations in visual and auditory feedback degrade learning and performance of every behavioral task so far examined. The findings have direct relevance to the human factors of human-computer interaction, in that the research itself made use of interactive computer- and video-based methods to deliberately introduce and evaluate the effects of such perturbations.

3. Although training improvements occur, this research has found no evidence to indicate that humans can adapt perfectly to spatial and/or temporal perturbations in sensory feedback which are not compliant with established motor feedback control mechanisms.

4. Because of noncompliant human-factors design features, sensory feedback perturbations arise routinely in the course of operational human interaction with telerobotic systems. Performance decrements even occur when video display viewing of a task is substituted for direct viewing.

5. A number of different modes and sources of sensory feedback perturbations can be identified which may potentially influence interactive human performance during space telerobot operation. Results from recent laboratory research using remote telerobot servicers show that manipulation tasks are adversely affected by both spatial perturbations and temporal delays, confirming in part the findings from earlier behavioral cybernetic research.

6. The human factors of workstation telepresence encompass the distinct social dynamic attributes of telerobotic systems, involving mutual sensory feedback exchange and control relationships among multiple human operators, computers, and telerobots. A social cybernetic model of human-computer interaction is described which uses social tracking concepts to characterize the sensory feedback control relationships among human and machine participants in complex, interactive systems. Current plans suggest that such social relationships will be an integral part of space station teleoperations, where station-, shuttle-, and ground-based systems may all be involved in mediating telerobotic tasks. One projection from the social cybernetic model is that a research emphasis on improving the capabilities of the interactive computer for controlling sensory feedback, from its own performance as well as from that of the human partner, will benefit the development of workstation telepresence.

REFERENCES

1. Purves, R.B., Lin, P.S., and Fisher, E.M., Jr., "A Plan for Time-Phased Incorporation of Automation and Robotics on the U.S. Space Station," SECOND CONFERENCE ON ARTIFICIAL INTELLIGENCE FOR SPACE APPLICATIONS, NASA Scientific and Technical Information Division, Washington, D.C., 1988, pp. 237-245.
2. Varsi, G., "Telerobotics for the Efficient Utilization of Space," PROCEEDINGS OF THE 39TH CONGRESS OF THE INTERNATIONAL ASTRONAUTICAL FEDERATION, International Astronautical Federation, Paris, 1988, IAF-88-023.
3. Varsi, G., and Herman, D.H., "Space Station as a Vital Focus for Advancing the Technologies of Automation and Robotics," SECOND CONFERENCE ON ARTIFICIAL INTELLIGENCE FOR SPACE APPLICATIONS, NASA Scientific and Technical Information Division, Washington, D.C., 1988, pp. 1-6.
4. Burke, J.D., "Lunar Resources: Evaluation and Development for Use in Lunar Base Program," PROCEEDINGS OF THE 39TH CONGRESS OF THE INTERNATIONAL ASTRONAUTICAL FEDERATION, International Astronautical Federation, Paris, 1988, IAF-88-586.
5. Burke, J.D., "Precursors and Adjuncts of a Lunar Base," PROCEEDINGS OF THE 39TH CONGRESS OF THE INTERNATIONAL ASTRONAUTICAL FEDERATION, International Astronautical Federation, Paris, 1988, IAF-88-616.
6. Podnieks, E.R., "Environmental Considerations for Lunar Base Engineering," ENGINEERING AND CONSTRUCTION AND OPERATIONS IN SPACE, Aerospace Division/ASCE, Albuquerque, NM, 1988, pp. 55-66.
7. Podnieks, E.R., and Roepke, W.W., "Mining for Lunar Base Support," LUNAR BASES AND SPACE ACTIVITIES OF THE 21ST CENTURY, W.W. Mendell (Ed.), Lunar and Planetary Institute, Houston, 1985, pp. 445-452.
8. Podnieks, E.R., and Roepke, W.W., "Mining Technology Requirements for Lunar Resource Development," PROCEEDINGS OF SYMPOSIUM '86 - THE FIRST LUNAR DEVELOPMENT SYMPOSIUM, Lunar Development Council, Pitman, NJ, 1987, pp. S86-38A to S86-38H.
9. NASA Advanced Technology Advisory Committee, "Advancing Automation and Robotics Technology for the U.S. Economy, Volumes I and II," NASA TM-87566, NASA, Washington, D.C., 1985.
10. Shaw, J.A., "Reducing Operator Error in Distributed Control Systems", INTECH, March, 1988, pp. 45-50.
11. Smith, K.U., and Smith, T.J., "Analysis of the Human Factors in Automation," SUCCESS FACTORS FOR IMPLEMENTING CHANGE - A MANUFACTURING VIEWPOINT, K.M. Blache (Ed.), Society of Manufacturing Engineers, Dearborn, MI, 1988, pp. 259-338.
12. Smith, T.J., and Smith, K.U., "Motor Feedback Control of Human Cognition - Implications for the Cognitive Interface," SOCIAL, ERGONOMIC AND STRESS ASPECTS OF WORK WITH COMPUTERS, G. Salvendy, S.L. Sauter, and J.J. Hurrell (Eds.), Elsevier, Amsterdam, 1987, pp. 239-254.
13. Smith, T.J., and Smith, K.U., "The Social Cybernetics of Human Interaction with Automated Systems," ERGONOMICS OF HYBRID AUTOMATED SYSTEMS I, W. Karwowski, H.R. Parsaei, and M.R. Wilhelm (Eds.), Elsevier, Amsterdam, 1988, pp. 691-711.
14. Smith, T.J., and Smith, K.U., "Behavioral Cybernetic Basis of Cognitive Performance. Experimental and Theoretical Analysis," ERGONOMICS (Special Issue on 'Methodological Issues in Cognitive Ergonomics'), 1989, In Press.
15. Collins, S.L., and Purves, R.B., "Remote Servicing of Space Systems," PROCEEDINGS OF THE SECOND CONFERENCE ON ARTIFICIAL INTELLIGENCE FOR SPACE APPLICATIONS, NASA,

Washington, D.C., 1988, pp. 523-535.
16. Keslowitz, S., "Open Control/Display System for a Telerobotics Work Station," PROCEEDINGS OF THE GODDARD CONFERENCE ON SPACE APPLICATIONS OF ARTIFICIAL INTELLIGENCE, NASA, Washington, D.C., 1987.
17. Mosher, R.S., "Industrial Manipulators," SCIENTIFIC AMERICAN, Vol. 211, No. 4, October, 1964, pp. 88-98.
18. Mosher, R.S., and Murphy, W., "Human Control Factors in Walking Machines," PROCEEDINGS OF AMERICAN SOCIETY OF MECHANICAL ENGINEERS HUMAN FACTORS CONFERENCE, Chicago, 1965.
19. Ruch, T.C., Patton, H.D., Woodbury, J.W., and Towe, A.L., NEUROPHYSIOLOGY, Saunders, Philadelphia, 1965.
20. Rahimi, M., "Human Factors Engineering and Safety in Robotics and Automation," HUMAN FACTORS SOCIETY BULLETIN, Vol. 30, No. 7, 1987, pp. 3-5.
21. Smith, K.U., HUMAN FACTORS ANALYSIS OF THE MECHANISM OF WALKING WITH SPECIAL REFERENCE TO DESIGN OF A PEDIPULATOR VEHICLE, University of Wisconsin Behavioral Cybernetics Laboratory, Madison, WI, 1965.
22. Smith, K.U., SENSORY-FEEDBACK ANALYSIS OF MANLIKE MACHINE SYSTEMS: A NEW BEHAVIORAL THEORY FOR HUMAN ENGINEERING, University of Wisconsin Behavioral Cybernetics Laboratory, Madison, WI, 1965.
23. Smith, K.U., DELAYED SENSORY FEEDBACK AND BEHAVIOR, Saunders, Philadelphia, 1962.
24. Smith, K.U., and Smith, W.M., PERCEPTION AND MOTION: AN ANALYSIS OF SPACE-STRUCTURED BEHAVIOR, Saunders, Philadelphia, 1962.
25. Smith, K.U., and Smith, M.F., CYBERNETIC PRINCIPLES OF LEARNING AND EDUCATIONAL DESIGN, Holt, Rinehart and Winston, New York, 1966.
26. Smith, K.U., "Cybernetic Psychology," THE PSYCHOMOTOR DOMAIN, R.N. Singer, Ed., Lea and Febiger, New York, 1972, pp. 285-348.
27. Smith, T.J., and Smith, K.U., "Cybernetic Factors in Motor Performance and Development," DIFFERING PERSPECTIVES IN MOTOR LEARNING, MEMORY, AND CONTROL, D. Goodman, R.B. Wilberg, and I.M. Franks, Eds., Elsevier, Amsterdam, 1985, pp. 239-283.
28. Smith, T.J., and Smith, K.U., "Feedback-Control Mechanisms of Human Behavior," HANDBOOK OF HUMAN FACTORS, G. Salvendy, Ed., Wiley, New York, 1987, pp. 251-293.
29. Smith, T.J., and Smith, K.U., "The Cybernetic Basis of Human Behavior and Performance," CONTINUING THE CONVERSATION. A NEWSLETTER OF IDEAS IN CYBERNETICS, Winter, 1988, Number 15, pp. 1-28.
30. Raibert, M.H., "Machines That Run," ROBOTICS AND MOTOR CONTROL: COMMONALITIES AND DIFFERENCES, Conference Paper No. 5, Simon Fraser University, Burnaby, B.C., Canada, June 3, 1987.
31. NASA Goddard Space Flight Center, "Space Station Flight Telerobotic Servicer Requirements," FTS REQUIREMENTS DOCUMENT FOR PHASE C/D, NASA RFP SS-GSFC-0043, Nov. 1, 1988, Section 3.0.
32. Smith, R.L., "Human Visual Requirements for Control and Monitoring of a Space Telerobot," ERGONOMICS OF HYBRID AUTOMATED SYSTEMS I, W. Karwowski, H.R. Parsaei, and M.R. Wilhelm (Eds.), Elsevier, Amsterdam, 1988, pp. 233-240.
33. Helmholtz, H.L.F. von, HANDBUCH DER PHYSIOLOGISCHEN OPTIK, 3 Volumes, Voss, Hamburg and Leipzig, 1856-1866.
34. Welch, R.B., PERCEPTUAL MODIFICATION. ADAPTING TO ALTERED SENSORY ENVIRONMENTS, Academic Press, New York, 1978.
35. Howard, I.P., and Templeton, W.B., HUMAN SPATIAL ORIENTATION, Wiley, New York, 1966.
36. Young, L.R., Oman, C.M., Watt, D.G.D., Money K.E., and Lichtenberg, B.K., "Spatial Orientation in Weightlessness and Readaptation to Earth's Gravity," LIFE SCIENCES, Vol. 225, 1984, pp. 205-208.
37. Vasyutin, V.V., and Tishchenko, A.A., "Space Coloristics," SCIENTIFIC AMERICAN, Vol. 261, No. 1, July 1989, pp. 84-90.
38. Huggins, C.T., Malone, T.B., and Shields, N.L., "Evaluation of Human Operator Visual Performance Capability for Teleoperator Missions," REMOTELY MANNED SYSTEMS, E. Heer (Ed.), California Institute of Technology, Pasadena, CA, 1973, pp. 337-350.
39. Chandlee, G.O., Smith, R.L., and Wheelwright, C.D., "Illumination Requirements for Operating a Space Remote Manipulator," ERGONOMICS OF HYBRID AUTOMATED SYSTEMS I, W. Karwowski, H.R. Parsaei, and M.R. Wilhelm (Eds.), Elsevier, Amsterdam, 1988, pp. 241-248.
40. Krishen, K., "Robotic Vision Technology and Algorithms for Space Applications," PROCEEDINGS OF THE 39TH CONGRESS OF THE INTERNATIONAL ASTRONAUTICAL FEDERATION, Paper No. IAF-88-028, International Astronautical Federation, Paris, 1988.
41. Krishen, K., "Space Robotic Vision System Technology," PROCEEDINGS OF ISA-88 INTERNATIONAL CONFERENCE, Paper No. 88-1615, ISA, Research Triangle Park, NC, 1988, pp. 1527-1537.
42. Smith, K.U., and Arndt, R., "Experimental Hybrid-Computer Automation in Perceptual-Motor and Social Behavioral Research," JOURNAL OF MOTOR BEHAVIOR, Vol. 1, No. 1, 1969, pp. 11-27.
43. Smith, K.U., and Smith, T.J., "Systems Theory of Therapeutic and Rehabilitative Learning with Television," JOURNAL OF NERVOUS AND MENTAL DISEASE, Vol. 148, 1969, pp. 386-429.
44. Smith, K.U., and Murphy, T.J., "Sensory Feedback Mechanisms of Handwriting Motions and Their Neurogeometric Bases," NEW HORIZONS FOR RESEARCH IN HANDWRITING, M. Herrick (Ed.), University of Wisconsin Press, Madison, WI, Chap. V, 1963, pp. 111- 157.
45. Smith, K.U., Cambria, R., and Steffan, J., "Sensory-Feedback Analysis of Reading," JOURNAL OF APPLIED PSYCHOLOGY, Vol. 48, No. 5, 1964, pp. 275-286.
46. Gould, J.D., and Smith, K.U., "Angular Displacement of Visual Feedback in Motion and Learning," PERCEPTUAL AND MOTOR SKILLS, Vol. 17, 1963, pp. 699-710.
47. Smith, K.U., and Ramana, D.S.V., "Characteristics of Head Motion with Variable Visual Feedback Delay, Velocity and Displacement," AMERICAN JOURNAL OF PHYSICAL MEDICINE, Vol. 48, No. 6, 1969, pp. 289-300.

48. Abbs, J.H., and Smith, K.U., Laterality Differences in the Auditory Feedback Control of Speech," JOURNAL OF SPEECH AND HEARING RESEARCH, Vol. 13, 1970, pp. 298-303.

49. Smith, K.U., and Putz, V., "Dynamic Motor Factors in Determining Effects of Retinal Image Feedback Reversal and Delay," AMERICAN JOURNAL OF OPTOMETRY AND ARCHIVES OF AMERICAN ACADEMY OF OPTOMETRY, Vol. 47, No. 5, 1970, pp. 372-383.

50. Sussman, H.M., and Smith, K.U., "Sensory Feedback Persistence in Determining Memory," JOURNAL OF APPLIED PSYCHOLOGY, Vol. 54, 1970, pp. 503-508.

51. Rothe, M., SOCIAL TRACKING IN CHILDREN AS A FUNCTION OF AGE, Masters Dissertation, Department of Psychology, University of Wisconsin - Madison, Madison, WI, 1973.

52. Lisberger, S.G., "The Neural Basis for Learning of Simple Motor Skills," SCIENCE, Vol. 242, 1988, pp. 728-735.

53. Lisberger, S.G., and Pavelko, T.A., "Brain Stem Neurons in Modified Pathways for Motor Learning in the Primate Vestibulo-Ocular Reflex," SCIENCE, Vol. 242, 1988, pp. 771-773.

54. Smith, K.U., and Greene, P., "A Critical Period in Maturation of Performance with Space-Displaced Vision," PERCEPTUAL AND MOTOR SKILLS, Vol. 17, 1963, pp. 627-639.

55. Smith, K.U., and Molitor, K., "Adaptation to Combined Reversal and Delay of Eyemovement-Retinal Feedback," JOURNAL OF MOTOR BEHAVIOR, Vol. 1, No. 4, 1969, pp. 296-308.

56. Smith, K.U., INDUSTRIAL SOCIAL CYBERNETICS, University of Wisconsin Behavioral Cybernetics Laboratory, Madison, WI, 1974.

57. Cherry, C., and Sayers, B. McA., "Experiments on Total Inhibition of Stammering by External Control and Some Clinical Results," JOURNAL OF PSYCHOSOMATIC RESEARCH, Vol. 1, 1956, pp. 233-246.

58. Sauter, S., and Smith, K.U., "Social Feedback: Quantitative Division of Labor in Social Interactions," JOURNAL OF CYBERNETICS, Vol. 1, No. 2, 1971, pp. 80-93.

59. Smith, K.U., and Kao, H., "Social Feedback: Determination of Social Learning," JOURNAL OF NERVOUS AND MENTAL DISEASE, Vol. 152, 1971, pp. 289-297.

60. Ting, T., Smith, M., and Smith, K.U., "Social Feedback Factors in Rehabilitative Processes and Learning," AMERICAN JOURNAL OF PHYSICAL MEDICINE, Vol. 51, 1972, pp. 86-101.

61. Smith, T.J., Smith, R.L., Stuart, M.A., Smith, S.T., and Smith, K.U. "Interactive Performance in Space - The Role of Perturbed Sensory Feedback," PROCEEDINGS OF THE 3RD INTERNATIONAL CONFERENCE ON HUMAN-COMPUTER INTERACTION, M. Smith (Ed.), Elsevier, Amsterdam, 1989, To be published.

62. Smith, R.L., THE EFFECTS OF TELEVISED ANGULAR DISPLACEMENT OF VISUAL FEEDBACK ON A REMOTE MANIPULATION TASK, Unpublished Masters Thesis, Texas A&M University, College Station, TX, 1986.

63. Stuart, M.A., and Smith, R.L., SPATIALLY DISPLACED VISUAL FEEDBACK AND PERFORMANCE OF FTS-LIKE TASKS, NASA Technical Report JSC-23547, NASA, Houston, TX, 1989.

64. Lawrence, D.A., Chapel, J.D., and Depkovich, T.M., "Issues, Concerns, and Initial Implementation Results for Space Based Telerobotic Control," PROCEEDINGS, 1987 GODDARD CONFERENCE ON SPACE APPLICATIONS OF ARTIFICIAL INTELLIGENCE (AI) AND ROBOTICS, NASA Goddard Space Flight Center, Greenbelt, MD, 1987, pp. 1-18.

65. Albus, J.S., Lumia, R., and McCain, H., Hierarchical Control of Intelligent Machines Applies to Space Station Telerobots," IEEE TRANSACTIONS ON AEROSPACE AND ELECTRON SYSTEMS, Vol. 24, No. 5, 1988, pp. 535-541.

66. McCain, H.G., and Andary, J.F., "The Flight Telerobotic Servicer Project and Systems Overview," PROCEEDINGS, 21ST ANNUAL ELECTRONICS AND AEROSPACE CONFERENCE, IEEE, Arlington, VA, 1988, pp. 97-102.

67. Smith, J.H., Gyamfi, M., Volkmer, K., and Zimmerman, W., "The Space Station Assembly Phase: System Design Trade-Offs for the Flight Telerobotic Servicer," PROCEEDINGS, 1988 GODDARD CONFERENCE ON SPACE APPLICATIONS OF ARTIFICIAL INTELLIGENCE (AI) AND ROBOTICS, NASA Goddard Space Flight Center, Greenbelt, MD, 1988.

68. Nicholas, J.M., "Interpersonal and group-behavior skills training for crews on space station," AVIATION, SPACE, AND ENVIRONMENTAL MEDICINE, Vol. 60, No. 6, 1989, pp. 603-608.

OVERVIEW OF NASA'S TECHNOLOGY DEVELOPMENT FOR HUMANS IN SPACE

Dr. J. Jenkins
NASA Headquarters

(Paper not provided by publication date.)

USAF SPACE HUMAN FACTORS OVERVIEW

Colonel D. Spoon
AFHSD/XAP, Brooks AFB

(Paper not provided by publication date.)

THE PARTICULATE ENVIRONMENT SURROUNDING SPACE STRUCTURES

B. Green
Physical Sciences, Inc.

M. Almadjian and C. Pike
Geophysical Laboratory, Hanscom AFB

(Paper not provided by publication date.)

SATELLITE MATERIAL CONTAMINANT OPTICAL PROPERTIES*

B. E. Wood, W. T. Bertrand, B. L. Seiber, and E. L. Kiech
Calspan Corporation Corporation/AEDC Operations
Arnold Engineering Development Center,
Arnold AFB, TN 37389

Capt. P. M. Falco
Air Force Wright Research and Development Center,
Wright-Patterson AFB, OH

and

Capt. J. D. Holt
Arnold Engineering Development Center, Arnold AFB, TN 37389

ABSTRACT

The Air Force Wright Research and Development Center and the Arnold Engineering Development Center are continuing a program for measuring optical effects of satellite material outgassing products on cryo-optic surfaces. This paper presents infrared (4000 to 700 cm^{-1}) transmittance data for contaminant films condensed on a 77 K geranium window. From the transmittance data, the contaminant film refractive and absorptive indices (n, k) were derived using an analytical thin-film interference model with a nonlinear least-squares algorithm. To date 19 materials have been studied with the optical contants determined for 13 of those. The materials include adhesives, paints, composites, films, and lubricants. This program is continuing and properties for other materials will be available in the future.

INTRODUCTION

Contamination control and prediction are becoming increasingly important as satellite applications become more sophisticated and anticipated satellite lifetimes are extended. A satellite mission can be terminated due to a cryogenically cooled optical system or sensor becoming contaminated. A spacecraft designer must predict effects of this contamination with a very limited amount of data. The Air Force Wright Research and Development Center (AFWRDC) is sponsoring a program to determine and predict effects of satellite material outgassing products on critical surfaces. This is being carried out through an experimental-analytical study of the optical properties of contaminants originating from satellite materials.

Infrared transmittance measurements of contaminants condensed from satellite material outgassing products have begun in the AEDC 2- by 3-ft chamber. The materials being investigated were heated to 125°C under vacuum and the outgassed products were frozen as thin films on a 77 K germanium window. A scanning Michelson-type interferometer was used to measure the transmittance over the 4000 to 450 cm^{-1} wavenumber range. These data were input into the TRNLIN computer code which was based on thin film interference with a germanium window as substrate. Using this program, the refractive (n) and absorptive (k) indices of the contaminant films were determined. Once determined, these n and k values can be used to calculate the transmittance or reflectance of other optical components contaminated with the outgassing products of the same satellite material and for any film thickness or radiation incidence angle. This is demonstrated through the use of another program, CALCRT, which can be used for calculating transmittances and/or reflectances of substrates with contaminant films present.

EXPERIMENTAL APPARATUS

Infrared transmittance measurements were made of satellite material outgassing contamination products on cryogenic surfaces in the AEDC 2- by 3-ft chamber (Fig. 1). The pumping system consisted of a turbomolecular pump with a mechanical forepump and a liquid nitrogen (LN_2) cooled chamber liner. The turbopump and the cryopanels were necessary to provide a near-contaminant-free vacuum. Base pressures before contamination deposition were generally in the mid-10^{-7}-torr range.

The outgassing products for contaminating the sample surface were generated using an effusion cell. It had a cylindrical aluminum body 8.89 cm (3.5 in.) long with an internal diameter of 4.45 cm (1.75 in.) The material to be analyzed was placed in the closed end of the effusion cell. Heating elements covered the outside cylinder surface, and the temperature of the cell was thermostatically controlled to 125°C. The evolved gases exited through the open end aperture, which was 3.81 cm (1.5 in.) in diameter.

The substrate surface on which the contaminants were deposited was a germanium window 6.99 cm (2.75 in.) square and was 4 mm (0.157 in.) thick. It was mounted in the center of the chamber and was cooled to near 77 K with a constant flow of LN_2. Germanium was picked for the deposition surface because it has good thermal conductance and a flat transmittance of 47 to 48percent over the 700 to 4000 cm^{-1} (2.5- to 14-μm) range. The mass flux deposited on the germanium was monitored using a quartz crystal microbalance (QCM) mounted adjacent to it. The QCM was also cooled with LN_2.

A commercial Michelson-type interferometer (See Fig. 1) was utilized in making infrared transmittance measurements of the deposited contaminant film on the germanium window. The interferometrically modulated infrared beam was collimated and passed through a KBr window on the vacuum chamber port,

*The research reported herein was sponsored by the Air Force Wright Research and Development Center (AFWRDC) and performed by the Arnold Engineering Development Center (AEDC), Air Force Systems Command. Work and analysis for this research were done by personnel of Calspan Corporation/AEDC Division, operating contractor for the AEDC aerospace flight dynamics facilities. Further reproduction is authorized to satisfy needs of the U.S. Government.

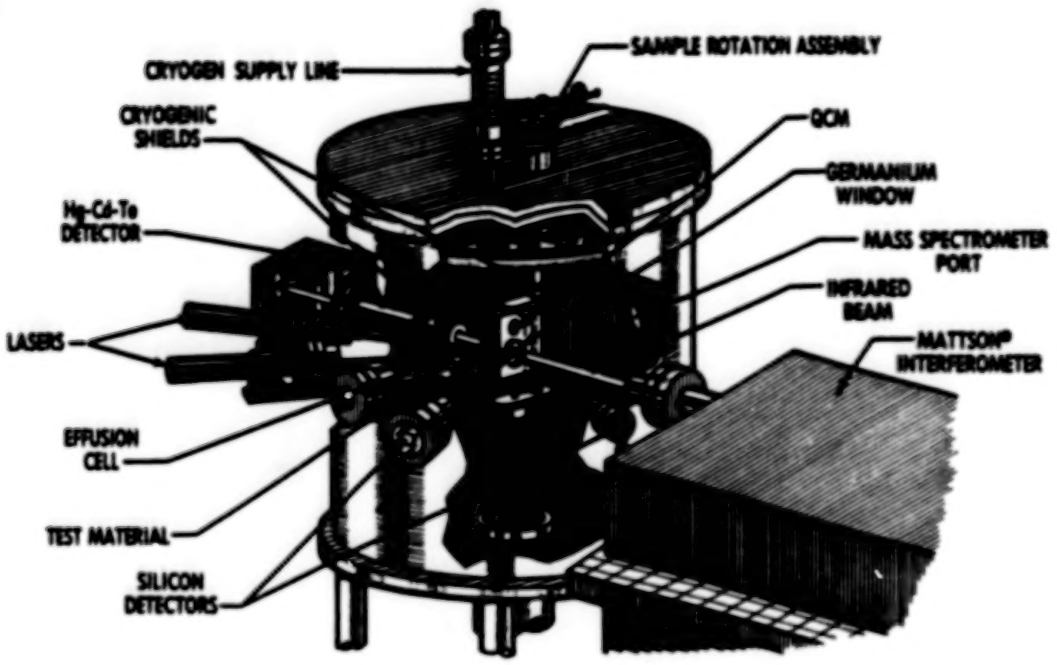


Fig. 1 Schematic of 2- by 3-ft cryogenic optics degradation chamber.

through the germanium window, through another KBr window on the opposite side of the chamber, and finally to detector optics and the Hg-Cd-Te detector. The wavenumber/wavelength range sensed was from 4000 to 450 cm^{-1} (2 to 22 μm). Typically, 32 scans were co-added for both the sample and reference measurements with a resolution of 2 cm^{-1}. A reference measurement was made before each sample measurement. Transmittance data were initially stored on the system hard disk and later transferred to flexible disks.

A list of some of the materials studied is given in Table I along with the condensed film properties measured. These include the refractive index at the He-Ne laser wavelength, film density, mass of material heated to 125°C, the total mass loss in percent (TML), and the maximum film thickness measured. The n's, k's column indicates whether or not the contaminant film was thick enough to determine the infrared n's and k's. Adhesives, paints, films, and graphite composites comprised the materials investigated. The adhesives were prepared by pouring them into an aluminum foil boat which was 3 by 1.5 by 1.5 in. (7.62 by 3.81 by 3.81 cm) and allowing them to cure. The empty aluminum foil boat was outgassed at 125°C for 24 hr prior to installation of the material. The paints were applied to aluminum foil strips and allowed to dry for the vendor-recommended time. All materials investigated were preconditioned by placing them in a 50-percent (±5 percent) relative humidity cell for 24 hr prior to installation in the chamber.

Table I. Materials investigated in AEDC 2 by 3 optical properties chamber

Material	Ref. Index .6328 μm	n's, k's Infrared	density, g/cc	Mass, grams	TML, %	Thickness, microns
RTV-732	1.44	yes	0.69	21.	4.53	2.75
DC93-500	1.44	yes	0.57	66.5806	.09	~1.3
DC6-1104	1.46	yes	0.80	109.7	--	9.28
Solithane	--	no	--	38.531	.20	~0.85
Kapton	1.21	yes	0.91	124.6644	1.26	3.59
S13G/LO	1.34	yes	1.11	53.4544	.16	2.96
RTV-566	1.43	yes	0.86	97.9884	.57	8.49
RTV-560	1.42	yes	0.9	78.0342	2.00	7.41
Mylar	1.28	yes	1.07	123.0593	.13	1.25
FEP teflon	--	no	--	142.7436	.01	~0.25
Stycast 2850	--	no	--	113.4790	.02	~0.37
Epoxi-patch	--	no	--	89.6960	[illegible]	~0.25
Crest 7450	1.32	yes	0.94	93.0990	.90	11.24
AS4/PEEK	--	no	--	82.9237	.03	0.25
AS4/J2	1.32	yes	0.85	110.7016	.24	3.01
EP30LI	1.32	yes	0.75	102.1996	.12	~1.5
AS4/PPS	--	no	--	93.1507	.03	--
Chemglaze Z306	1.45	yes	1.03	6.7162	2.07	~1.82
Chemglaze A276	1.47	yes	1.00	14.9695	4.80	~10.7

EXPERIMENTAL TECHNIQUES

In order to make accurate n, k measurements, the transmittance must be measured for carefully determined film thicknesses. The thin film interference technique provides an excellent method for calculating these discrete thicknesses. This technique has been described previously[1,2] and will only be reviewed here. As a thin film forms on a reflecting substrate the intensity of a reflected beam of radiation (such as from a laser) will vary sinusoidally. From thin interference equations, the maxima and minima locations can be used to accurately calculate the film thickness at these maxima-minima locations.[1] In order to make these calculations, the film refractive index, n, must be known for the incident radiation wavelength. To determine n, two He-Ne laser beams were incident on the germanium window at two angles (24.0 and 67.5 deg). The reflected beams were detected with silicon solar cells. Interference maxima were observed on a strip chart recorder as the contaminant film deposited. The refractive index at 0.6328 μm was determined from the interference patterns by using the fringe (interference maxima or minima) counting technique described in Refs. 1 and 2. Knowing the refractive index at 0.6328 μm, the film thicknesses for the reflected interference maxima and minima were calculated from the thin film interference equation,

$$t = m\lambda / 2n\left(1 - \frac{\sin^2\theta}{n^2}\right)^{0.5}$$

where t = film thickness (μm), λ = wavelength (μm), n = real part of refractive index at wavelength λ, θ = incidence angle (deg), and m = order of interference where m = 1,2,3, etc. for maxima, and 1/2, 3/2, 5/2 etc. for minima. The refractive indices determined at the wavelength of 0.6328 μm are shown in Table 1 for outgassing products of the materials analyzed. Knowing the film thickness and the mass condensed (determined using the QCM), the film density also was calculated.

CRYOGENIC CONTAMINATION CHAMBER EXPERIMENTAL PROCEDURES

After the sample material had been preconditioned, it was inserted in the effusion cell and installed in the 2- by 3-ft chamber. The centers of the germanium window and the QCM were aligned equidistant from the effusion cell centerline so that the two surfaces would see the same flux rate. Pumpdown of the chamber then began. After the pressure had reached 10^{-5} torr, LN_2 cooling of the chamber liner, germanium window, and QCM began. Both QCM and germanium window reached thermal equilibrium before measurements began. At this point the chamber pressure was in the mid-to-high 10^{-7}-torr range. An initial germanium transmittance measurement was made to insure the window was clean. The effusion cell warmup to 125°C began and the laser-solar cell outputs were observed with time on a strip chart recorder. Outgassed components were condensed on the germanium window and the QCM. As the outgassed products condensed on the germanium window, the thin film interference caused the laser-solar cell outputs to exhibit sinusoidally varying signals. Deposition continued until the first interference minimum (quarter wavelength) was reached. The transmittance of the germanium window with the deposited film was then measured. The QCM change in frequency with time was also recorded during contaminant deposition.

Once the transmittance measurements were completed, the germanium was rotated back into the original deposition position, and the film buildup and transmittance measurements continued. This procedure was repeated for as many thicknesses as could be obtained before the deposition rate decreased to a minimal value. For some materials, films up to 25 interference maxima thick were obtained, whereas for others only 1 to 2 were observed. Transmittance measurements were made for as many thicknesses as possible to maximize the accuracy of the n,k calculations. In some cases transmittance measurements were made during warmup of the germanium window. This helped to determine the temperature where individual contaminant species were re-evaporated and to aid in their identification.[2] The total mass loss percentage (TML) was determined by dividing the mass lost due to outgassing for 24 hr (at 125°C under vacuum) by the original mass.

EXPERIMENTAL DATA

All of the materials listed in Table 1 have been analyzed. In some instances, not enough contaminant was obtained to determine the infrared n's and k's. Generally, film thicknesses of about 1.5 μm or greater were required for optical properties determination. Examples of the infrared transmittance data obtained are shown in Figs. 2 through 4 for Mylar® film (5 mil), AS4/J2 thermoplastic composite, and Chemglaze® Z306 black paint, respectively. These figures show the infrared transmittance of the 77 K germanium window with three film thicknesses of contaminant condensed as material outgassing products. Infrared transmittance spectra for contaminant films from some of the other materials studied are reported in Ref. 2.

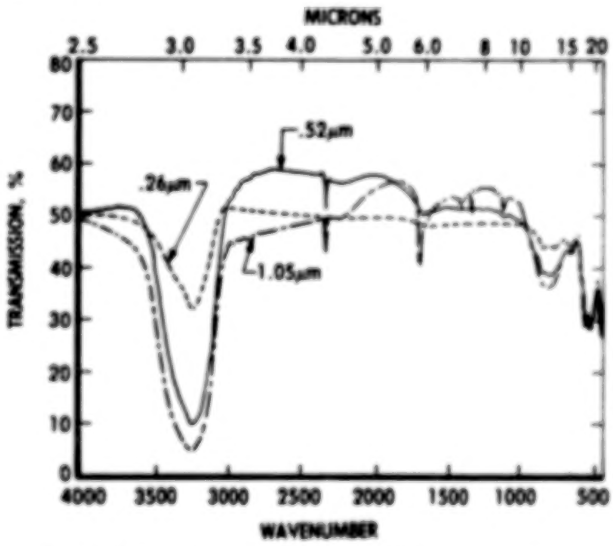


Fig. 2. Transmittance of condensed Mylar components-0.26, 0.52, and 1.05 μm thick.

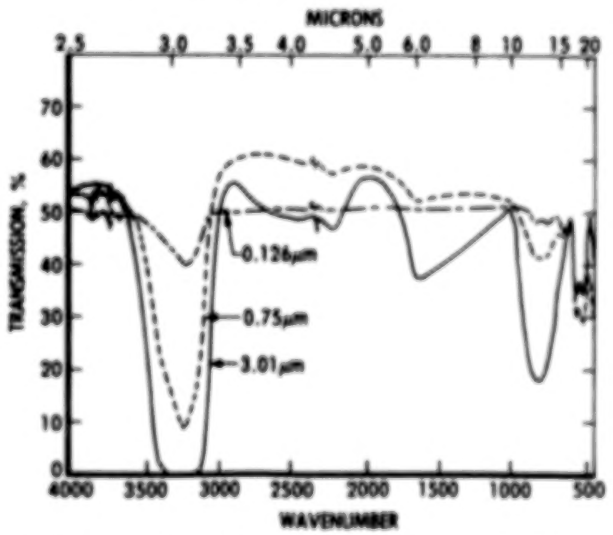


Fig. 3. Transmittance of condensed AS4/J2 thermoplastic composite components 0.126, 0.75, and 3.01 μm thick.

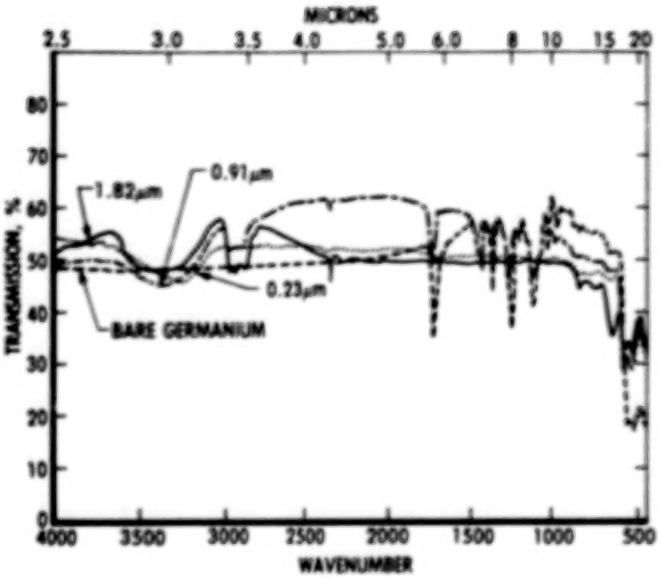


Fig. 4. Transmittance of condensed Chemglaze Z306 components 0.23, 0.91, 1nd 1.82 μm thick.

Fig. 2 shows the infrared transmittance of the germanium window with condensed Mylar contaminant films of 0.26, 0.52, and 1.05 μm. (Bare germanium transmits about 49 percent between 4000 and 700 cm^{-1}.) The major identifiable component outgassed from Mylar is water as evidenced by the broad strong absorption band (See Fig. 2) centered near 3300 cm^{-1}. A trace of CO_2 shows up as the thin spike located at 2340 cm^{-1}. For the 1.05-μm-thick film, several weak bands show up between 900 and 1500 cm^{-1}, which are probably due to hydrocarbons. Total mass loss (TML) was found to be 0.13 percent.

For the AS4/J2 composite material, condensed outgassing products of 0.126, 0.75, and 3.01 μm produced transmittance spectra shown in Fig. 4. The spectra show a strong resemblance to that observed for Mylar. Water vapor again was the dominant species outgassed with even less CO_2 observed than for the Mylar. The water bands are more pronounced as thicker contaminant films were obtained for the composite material than for Mylar. The absorption band locations 3300, 2250, 1600, and 800 cm^{-1} are all due to water. There was no evidence of other outgassed species other than the previously mentioned trace of CO_2. The TML found for AS4/J2 was 0.24 percent.

For Chemglaze Z306 (Fig. 4), transmittance spectra of films 0.23, 0.91, and 1.82 μm are shown. The painted samples were allowed to dry for 7 days prior to the measurements. The outgassing species appear to be predominantly hydrocarbons, with some water and a trace of CO_2. The hydrocarbons are identified by the bands near 3000 cm^{-1} and several between 1800 and 800 cm^{-1}. The TML observed was 2.07 percent.

The housekeeping data were monitored and stored by a computer. A typical set of data included the QCM parameters (frequency, mass, mass rate, and temperature), the laser-solar cell outputs, the effusion cell temperature, and the germanium window temperature. Typical plots of these parameters are contained in Ref. 2.

OPTICAL PROPERTIES DETERMINATION

Most contaminant problems encountered in space involve contaminant thicknesses of a few micrometers or less. Therefore, thin film interference equations are generally used to predict contaminant effects on the reflectance or transmittance of an optical element. These calculations require knowledge of the optical properties of the contaminant film, the refractive and absorptive indices, n and k. They are components of the more general expression for the complex refractive index given by $n^* = n - ik$. In order to determine the complex refractive index of the thin, solid-contaminant films, an analytical model called TRNLIN has been developed.[3,4] The model uses the expressions given in Ref. 5 for thin film transmittance and reflectance. The model assumes the germanium window is a thick film, and all interference occurs within the thin contaminant film. The expressions derived were for the normal transmittance of a thin film on a nonabsorbing substrate, which accurately represents the experimental conditions under which the transmittance measurements were made. Transmittance values of the germanium window with known contaminant film thicknesses were input into the TRNLIN program. The program uses a nonlinear least-squares convergence routine for determining n and k. The n's and k's were determined at every 2 cm^{-1} in the range from 700 to 4000 cm^{-1}. The refractive and absorptive indices determined from the transmittance data are shown in Figs. 5 through 7 for Mylar film, AS4/J2 thermoplastic composite, and Chemglaze Z306 black paint, respectively. The refractive indices are shown at the top with the absorptive indices directly below.

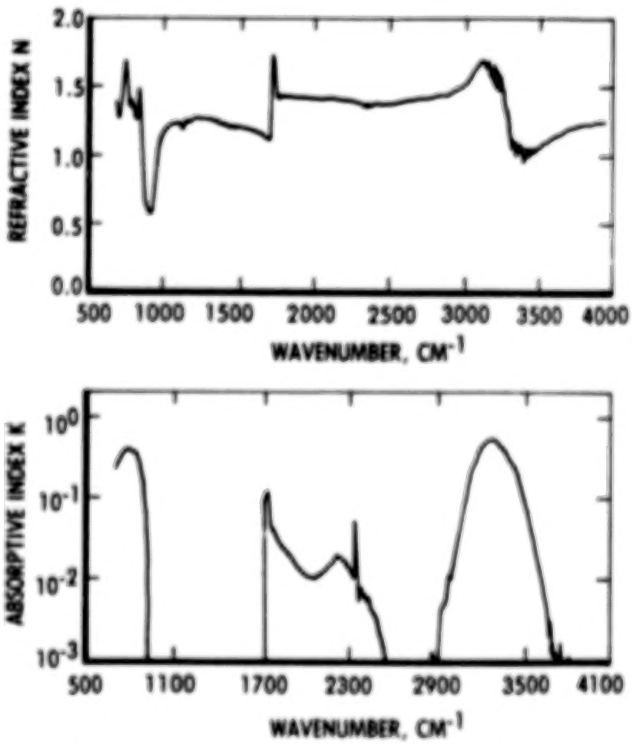


Fig. 5. Refractive and absorptive indices for Mylar film (5 mil) outgassing products.

The standard deviations for each wavenumber were calculated as part of the TRNLIN program. They generally varied with wavenumber, but for the most part were on the order of 0.01 for the refractive index and 0.001 for the absorptive index. Tabulated n, k data for all of the materials discussed will eventually be included in an AFWRC data base.

TRANSMITTANCE AND REFLECTANCE CALCULATIONS USING CALCRT

To realize the maximum utility of the n,k data generated from the experimental and analytical studies, a computer program, CALCRT was developed. CALCRT, written in FORTRAN IV, calculates the transmittances and reflectances for a radiation

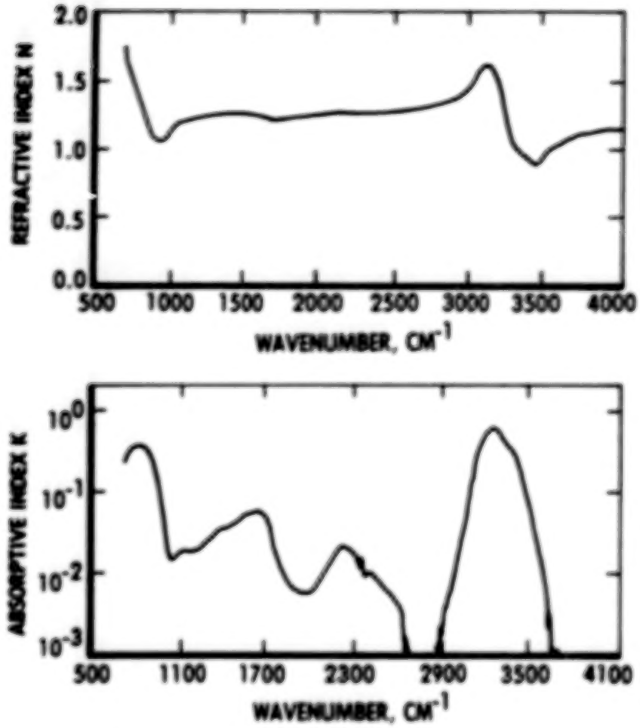


Fig. 6. Refractive and absorptive indices for AS4/J2 thermoplastic composite outgassing products.

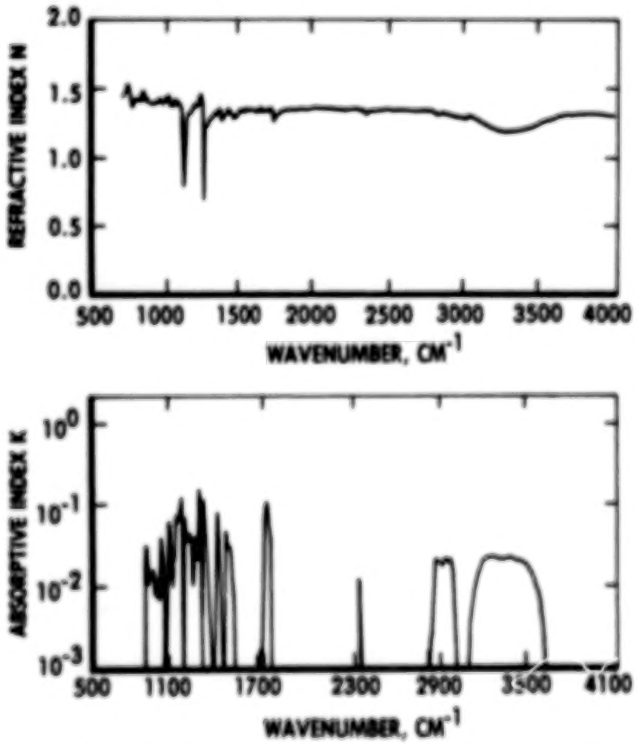


Fig. 7. Refractive and absorptive indices for Chemglaze Z306 outgassing products.

beam that strikes a film and substrate system which has planar interfaces that are, ideally, infinite in extent. The film and substrate are sandwiched between semi-infinite vacuums, and the refractive indices on each side of the filmsubstrate are identically equal to one.

The user must supply the optical constants and thicknesses of the film and substrate, and the incidence angle of the beam. The user must choose either an output format that gives the reflectances and transmittances as functions of wavenumber or wavelength at a constant film thickness, or an output format that displays the transmittances and reflectances versus film thickness at a constant wavenumber or wavelength.

To show how CALCRT can be used, an example of transmittance versus wavenumber was calculated using the n's and k's previously determined for Chemglaze Z306 material. Figure 8 shows curves of transmittance versus wavenumber for a 1.82-μm-thick contaminant film. The two curves overplotted are the actual measured spectral transmittance and the calculated transmittance curve based on the n's and k's determined. As seen in Fig. 8, there is excellent agreement between the experimental and calculated curves. Differences of about 3 percent were the largest seen.

CALCRT was also used to calculate the transmittance dependence on film thickness. An example is shown in Fig. 9 for contaminant films deposited from the outgassing products of a composite material AS4/J2, which is a thermoplastic. The three curves shown are for (top to bottom) 2000, 800, and 3250 cm^{-1}. The corresponding n,k values used for the calculations are also given in Fig. 9. There is relatively little absorption observed for 2000 cm^{-1} ($k = 0.0061$) and only the interference phenomena is seen superimposed on the germanium transmittance. For the 3250 cm^{-1} curve, however, the strong absorption index ($k = 0.597$) reduces the film-substrate transmittance to near zero for a film thickness of approximately 1.5-μm thick. (The high absorption index at 3250 cm^{-1} is typical of films containing water). The middle curve (for 800 cm^{-1}) is for a region where there is medium absorption ($k = 0.334$) and represents the location of another water absorption band (See Fig. 3). The solid curves show the analytical results calculated using CALCRT and the derived n's and k's, whereas the symbols denote the actual experimental values. As can be seen in Fig. 9, the analytical and experimental results agree very well. The important point that should be re-emphasized about the n's and k's of contaminant films is that, once determined, they can be used to calculate transmittances and reflectances for any desired film thickness, incidence angle, or substrate (provided the substrate refractive index is known).

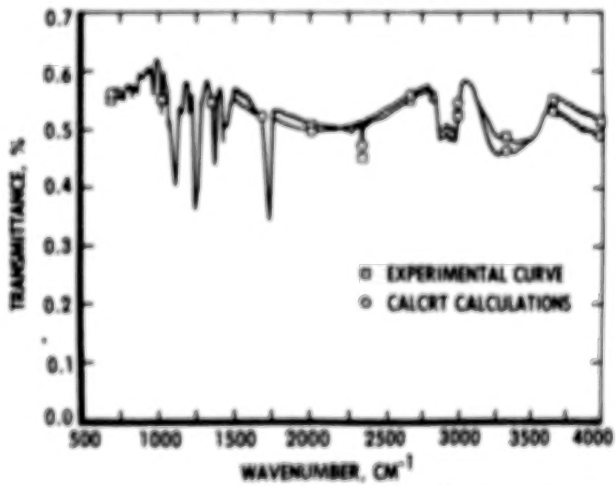


Fig. 8. Transmittance versus wavenumber for experimental measurement and CALCRT calculations for a 1.82-μm-thick Chemglaze Z306 outgassing films for 2000, 800, and 3250 cm^{-1}.

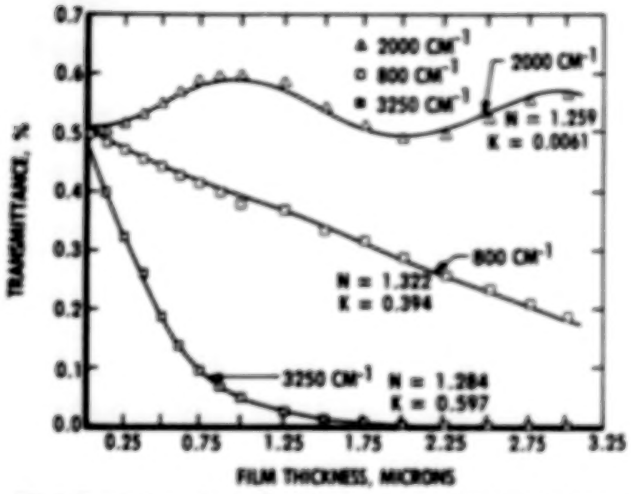


Fig. 9. Comparison of calculated and experimental transmittance values versus film thickness for AS4/J2 outgassing products contaminant films for 2000, 800, and 3250 cm^{-1}.

SUMMARY

The Air Force Wright Research and Development Center (AFWRDC) and the Arnold Engineering Development Center (AEDC) have initiated a program for measuring contaminant surface effects of satellite material outgassing products on cryo-optic surfaces. The complex refractive index components, n and k, of thin contaminant films condensed on a cryogenic surface, were determined from experimental infrared transmittance measurements for the wavenumber range from 4000 to 700 cm^{-1}. The materials studied were heated to 125°C and outgassing products were condensed on a 77 K germanium window. From the infrared spectra, outgassing components can be identified by absorption band locations. Water, CO_2, hydrocarbons, and silicones are easily identified. The n's and k's were determined using a thin film interference analytical model and a nonlinear least-squares algorithm. Infrared transmittance data and the n's and k's determined, are presented for 5-mil-thick Mylar film, AS4/J2 composite material, and Chemglaze Z306 black paint. These are only 3 of approximately 20 materials which have been investigated. For information on other materials, see Ref. 2.

A computer program, CALCRT, has been written which calculates the transmittance and reflectance values for the following parameters: substrate refractive index and thickness, contaminant refractive index and film thickness, incidence angle, wavenumber, and wavelength. This provides a potential optical property user with the program for utilizing the n's and k's generated for the materials mentioned previously.

REFERENCES

1. Wood, B. E. and Smith, A. M., "Infrared Reflectance of Condensed Gas Films," *AIAA Progress in Astronautics and Aeronautics: Thermophysics and Thermal Control*, Vol. 65, Ed. Raymond Viskanta, 1979, pp. 22-38.

2. Wood, B. E., Bertrand, W. T., Bryson, R. J., Seiber, B. L., Palco, P. M., and Cull, R. A., "Surface Effects of Satellite Material Outgassing Products," *Journal of Thermophysics and Heat Transfer*, Vol. 2, No. 4, October 1988, pp. 289-295.

3. Wood, B. E. and Roux, J. A., "Infrared Optical Properties of Thin H_2O, NH_3, and CO_2 Cryofilms," *Journal of the Optical Society of America*, Vol. 72, No. 6, June 1982, pp. 720-728.

4. Wood, B. E. and Roux, J. A., "Infrared Optical Properties of Thin CO, NO, CH_4, HCl, N_2O, O_2, N_2, and Ar Cryofilms," *Progress in Astronautics and Aeronautics, Spacecraft Contamination: Sources and Prevention*, Vol. 91, Ed. J. A. Roux and T. D. McCay, 1984, pp. 139-161.

5. Heavens, O. S., *Optical Properties of Thin Films*, Dover Publications, Inc., New York, 1955.

EXPERIMENTAL INVESTIGATIONS OF LOW-ENERGY (4-40 eV) COLLISIONS OF $O^-(^2P)$ IONS AND $O(^3P)$ ATOMS WITH SURFACES

A. Chutjian,* O. J. Orient* and E. Murad+

*Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109
+Air Force Geophysics Laboratory, Hanscom Air Force Base, MA 01731-5000

ABSTRACT

Using a newly-developed, magnetically confined source low-energy, ground state oxygen negative ions and neutral atoms are generated. The energy range is variable, and atom and neutrals have been generated at energies varying from 2 eV to 40 eV and higher. We find that the interaction of these low-energy species with a solid magnesium fluoride target leads to optical emissions in the (at least) visible and infrared regions of the spectrum. We will discuss briefly details of the photodetachment source. We will then present spectra of the neutral and ion "glows" in the wavelength range 250-850 nm (for O^-) and 600-850 nm (for O), and discuss the variability of the emissions for incident energies between 4 and 40 eV.

1 INTRODUCTION

The problem of vehicle "glow" emission in low-earth orbit (LEO) has been documented in some detail over the past several years. [See Ref.1 for a recent review of missions and observations.] By way of summary, the optical emissions observed from the the Shuttle bay window are from Shuttle surfaces oriented in the "ram" direction (along the Shuttle velocity vector), and have been associated with interaction of 5 eV ground-state oxygen atoms with the various surface materials. The emission spectrum in LEO has been measured, at the 3.4 nm resolution level, and found to be continuous in the range 450-800 nm, with a broad maximum at about 680 nm. Higher resolution spectra in LEO, extending to shorter wavelengths, are not available and would clearly be helpful in revealing any latent band structure and in identifying the emitting species.

There is thus far no definite explanation for the glow process. However, various scenarios have been suggested to account for the glow, with the most promising being that of Swenson, *et al* [2] in which the emission has been proposed to arise from the NO_2 recombination continuum. The excited NO_2^* molecules are formed on the surface by successive surface-mediated reactions of N and O by the so-called Langmuir-Hinshelwood mechanism. Desorbed, excited NO_2^* molecules emit in the $\hat{A}^2B_1 \rightarrow \hat{X}^2A_1$ continuum. The lifetime of the $\hat{A}$ state (which determines the spatial extent of the glow above the Shuttle skin), and the wavelengths of emission are both consistent with Shuttle observations.

Because of the difficulty in generating fast, ground-state atomic oxygen in the laboratory, simulation followed by a clear understanding of the mechanism for the surface glow has been a tedious process. Methods for generating oxygen atoms usually suffer from one or more of the following properties: the atoms are too slow – less than 1 eV in energy (plasma discharges); there is an unknown admixture of excited molecules and atoms (positive-ion charge exchange, or high-power laser sources that use either a solid target or a dense gaseous target); or peak pulsed O-atom fluxes are so high (order of 10^{30} atoms/cm^3-sec for so-called "accelerated testing" by high-power pulsed laser sources) that one "burns plastic" rather than bathing the surface in an LEO-encountered flux of 10^{13}-10^{15} atoms/cm^3-sec.

We present in Sec. 2 a description of the JPL atomic oxygen source which generates ground-state O-atoms of quantum state, energy, flux, and beam purity simulating that encountered in LEO. It has been used to obtain first spectra of the glow phenomenon in the laboratory. The MgF_2 surface chemiluminescence spectra using low-energy O^- ions is described in Sec. 3, and chemiluminescence spectra using low-energy O-atoms are given in Sec. 4.

2 OXYGEN-ATOM SOURCE DESCRIPTION

The JPL atomic oxygen source was designed specifically to study the mechanism of the spacecraft glow phenomenon, degradation of materials in LEO, and to carry out basic fast-atom collision studies with neutral gaseous targets. Techniques used in the source involve the following steps carried out in a uniform, high-intensity solenoidal magnetic field (see Fig. 1): (a) generate $O^-(^3P)$ ions via dissociative attachment to NO at 8 eV electron energy (point G); (b) accelerate the O^- ions and electrons to the desired final energy (5 eV, say); (c) by trochoidal deflection (T_1) separate the higher-velocity electrons from the slower O^- ions, and trap the electrons in their Faraday cup; (d) detach the electron from O^- by a cw laser and a multiple-pass mirror (M) geometry; and (e) direct the O^- and O beams toward the target, and reflect the undetached O^- beam by biasing the target negative with respect to the O^- kinetic energy. Alternatively, a second trochoidal deflector, located immediately after the mirrors M, was sometimes used to separate the ground-state $O(^3P)$ atoms from undetached O^- ions, and trap the latter in a separate Faraday cup for analysis of the O^- beam.

The laser wavelengths are restricted to visible lines of an argon-ion laser so that

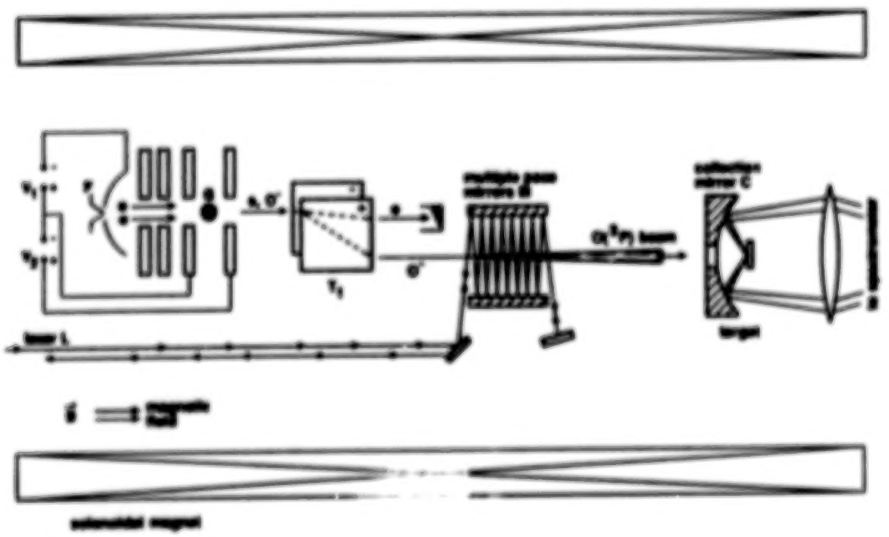


Figure 1. Schematic diagram of the magnetically-confined, photodetachment O-atom source.

detachment results exclusively in ground-state atoms [3]. For a 20 watt laser, 5 eV ions, and 100 passes across the confined ion beam, one obtains about a 15% detachment efficiency into $O(^3P)$. Furthermore, if wavelengths shorter than about 360 nm are used, one may also generate a mixture of $O(^1D)$ and $O(^3P)$ atoms. This capability is useful in studying differential reaction effects of electronically-excited atoms, such as O-atom addition to double-bonds in epoxies and organic molecules.

In order to minimise rapid divergence of the O-atom beam after detachment, care was taken in the source to restrict the energy of the O^- ions perpendicular to the B field. This involves choosing the target gas (at G) to provide small O^- kinetic energy at onset, and giving due regard to the space-charge repulsion between the ions, especially at high O^- currents.

For both the O^- and O-atom studies, the same sample of MgF_2 was used as the target. A 95% transmitting tungsten gauze covered the surface to prevent surface charging by the O^- beam; and also to allow repulsion of the O^- beam during the O-atom measurements. A large, high-reflectivity mirror at C (Fig. 1) was used to collect a broad spatial extent of optical emissions from the target. These emissions were focused onto the entrance plane of a fast f/3.5 double-grating monochromator capable (for the O-atom case) of attenuating by a factor 10^9 laser lines at 500 nm from spectral emissions at 600 nm. The detection system used an RCA phototube with a gallium arsenide photocathode and a manufacturer's stated long-wavelength cutoff of 900 nm. Fast, pulse-counting electronics were used, and spectra accumulated by multichannel scaling.

3 THE NEGATIVE ION [$O^-(^3P)$] GLOW

We show in Fig. 2 spectra of the $O^-(^3P)$ glow from a MgF_2 surface in the wavelength range 250-850 nm, and at four O^- energies ranging from 5-40 eV. The currents were in the range 1.0 -5.0 μa, or fluxes of (0.32 -1.6) $\times$ 10^{14} ions/cm^2-sec. The spectra in Fig. 2 were normalized to a common flux of 1.0 $\times$ 10^{14}. Care was taken in these measurements to prevent surface charging of the dielectric target by use of a transparent, conducting tungsten gauze on the target surface. This gauze itself was shown not to contribute to

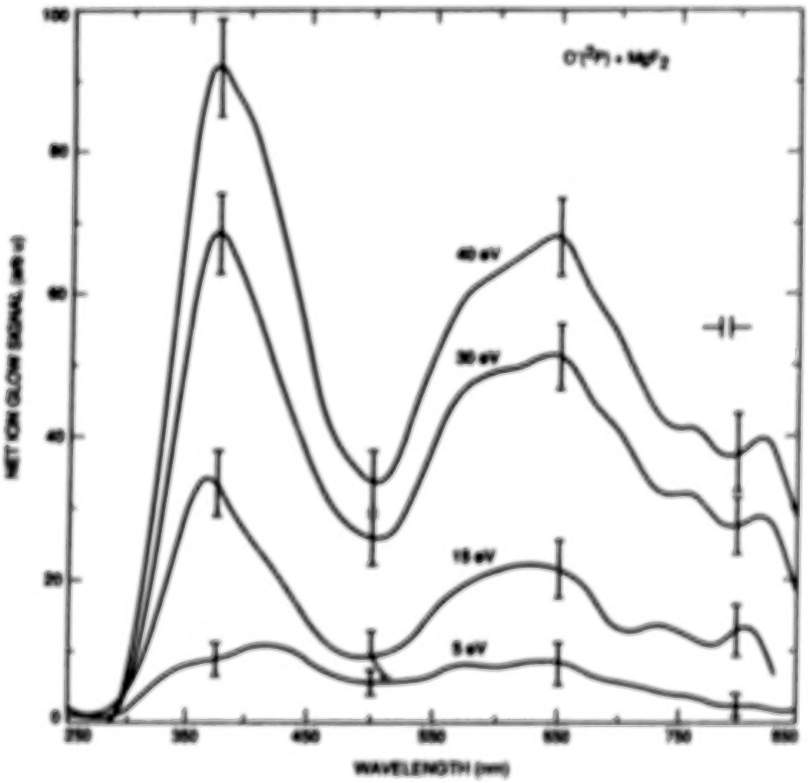


Figure 2. Glow signal of negative oxygen ions $O^-(^3P)$ from a MgF_2 surface, at the indicated O^- collision energies.

the observed spectra. Also, a glow contribution arising from interaction of the O^- beam with the background gas (mainly NO) was observed. This contribution to the spectrum was measured by taking spectra with the tungsten gauze biased negatively, so that the O^- beam was reflected just above the MgF_2 target.

One sees in Fig. 2 that the O^- glow from MgF_2 is characterized by two broad, unstructured (at the 10 nm resolution level of the present measurements) spectral emissions: one peaked near 375 nm, and the second at 600-650 nm. The spectral intensity is a strong function of O^- energy, with emission being strongest at 40 eV, and diminishing rapidly at energies near 5 eV. No attempts have been made to indentify the origin of these features pending experiments to be carried out under more controlled vacuum and surface conditions (see Sec. 4).

It is interesting to note that at high fluxes (order of 10^{14}), the O^- glow is easily visible to the eye, having a bluish appearance (not resembling, that is, the orangish Shuttle glow!).

4 THE ATOMIC OXYGEN [$O(^3P)$] GLOW

Shown in Fig. 3 are spectra in the wavelength range 600-850 nm arising from the interaction of 4, 5 and 40 eV $O(^3P)$ oxygen atoms with the same MgF_2 surface. Comparison of these

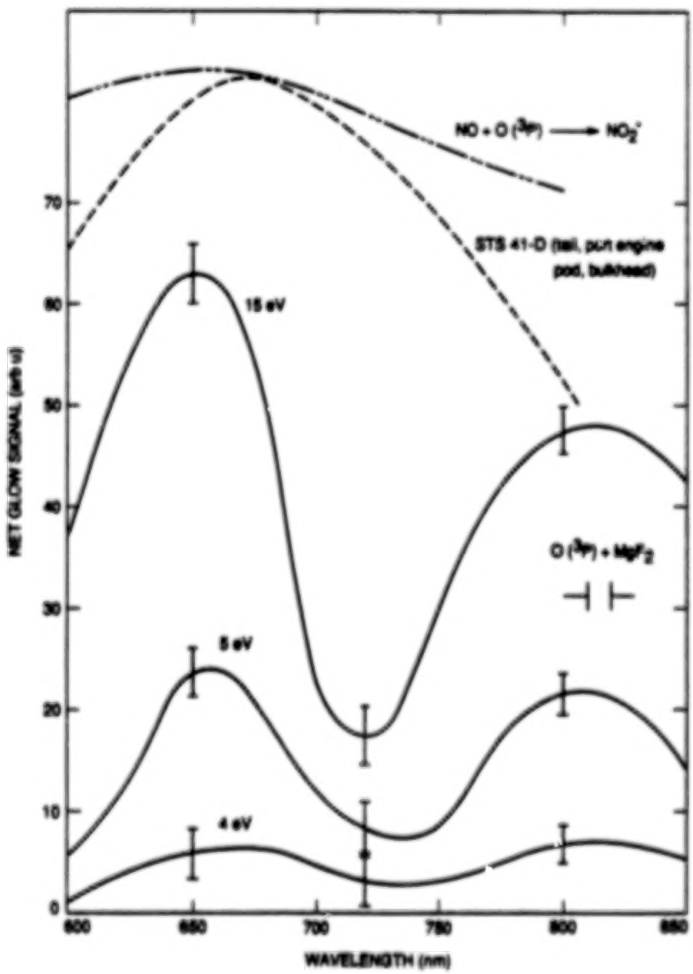


Figure 3. Glow signal of ground state atomic oxygen atoms $O(^3P)$ from a MgF_2 surface, at the indicated O-atom collision energies.

spectra are made with glow results from the STS 41-D mission [2], and from laboratory-measured NO + O recombination spectra at thermal oxygen-atom energies [4,5].

The spectra in Fig. 3 were approximately two-to-three orders of magnitude weaker than the O^- glow spectra of Fig. 2. Reasons for this are (a) the efficiency of the photodetachment step is only about 10%, (b) the O-beam, no longer confined by the magnetic field, diverges as it travels toward the target from a combination of imparted kinetic energy in the dissociative attachment step, and effects of space-charge repulsion in the O^- beam, and (c) the conversion efficiency of O-atoms to visible photons is estimated to be quite low, about 2.5×10^{-6} [6].

As a result, a number of other background sources had to be identified and eliminated from these spectra. The major contributors were : (a) a persistent fluorescence signal B_L arising from the detachment mirrors with the laser on, (b) photons B_{O^-} arising from interaction of the O^- beam with the background gas, and (c) photons B_F from the electron-gun filament. The net glow signal G was obtained over four cycles of signal-gathering. These were (1) photon counts with laser on and O^- beam on, (2) counts with laser on and O^- off, (3) counts with laser off and O^- on, and (4) counts with laser off and O^- off. The net glow signal G was then given by (1) - (2) - (3) + (4). The error limits shown in Fig. 3 reflect the sum of the four statistical errors in the individual counting cycles.

Also considered in this sequence was the fact that the background B_{O^-} in steps (1) and (3) are not the same, due to depletion of O^- by detachment. We found that this depletion had to be about a factor of 2.5 greater than could be accounted for solely by detachment. The reason for this is not clear. It could lie either in the measured value [3] of the detachment cross section being a factor of 2.5 too low; or that the detachment cross section of O^- is enhanced (by the same factor) via Landau resonances of the detached electron in the solenoidal magnetic field. Calculations based on works of Crawford [7] and observations by Krause [8] show that these resonances do persist up to 1 eV above detachment threshold. Enhancement is thus possible in our case, given the homogeneity of magnetic field (estimated to be better than 0.5% in the detachment region), laser bandwidth (8 GHz), and spacing of the resonances (169 GHz, neglecting overlapping fine-structure transitions and Zeeman splittings).

One striking feature in Fig. 3 is that the glow is continuous (10 nm resolution level) and shows two prominent maxima, one at 650 nm and the second at 800-825 nm. The first maximum is in good agreement with the STS 41-D data and laboratory recombination spectra. No evidence is seen for a second maximum in these last data. However, the NO_2 recombination continuum may take on a different shape, depending on the recombining species and the available internal energy; and on the type of Shuttle surface material and its temperature. It is interesting to note recent laboratory glow results wherein a single maximum at 820 nm in glow signal was seen in an O (0.16 eV energy) + NO + Ni surface experiment [9].

The second striking feature is that the glow signal falls below our present detection limit at an energy of 4 eV, and this represents an upper energy bound to the true threshold. To obtain an estimate of the threshold, we use the following pieces of information: (a) the O + NO (surface) system gains 3.1 eV by the ON-O bond formation, (b) one requires about 0.5 eV to dislodge the electronically-excited NO_2^* from the surface, (c) the NO_2^* leaves the surface with about 1.3 eV kinetic energy, and (d) the maximum internal electronic excitation energy of NO_2^* corresponds to the 400 nm (3.1 eV) onset of the LEO-observed glow. Thus, the center-of-mass threshold energy T_{cm} is given by T_{cm} + (a) = (b) + (c) + (d), or T_{cm} = 1.8 eV, and T_{lab} = 1.8 x (16/10.4) = 2.8 eV (neglecting the small thermal energy of the surface-bonded NO). This is consistent with our upper limit of 4 eV.

As in the O^- case, no identification of the emission continua is possible without understanding the mechanism of the glow process. To this end, work is currently underway to carry out the glow measurements under ultrahigh vacuum conditions, by playing different gases over the solid target surface and identifying which gases, if any, enhance the glow.

Work is also planned to explore the spectral region below 600 nm. While some of this region is obscured by the detaching argon-ion laser lines, wavelengths below 450 nm are accessible.

5 ACKNOWLEDGEMENTS

We thank Mr. A. Timpe for his assistance in machining many of the components of the source. This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was supported by the Air Force Geophysics Laboratory and DARPA through agreement with NASA.

6 REFERENCES

1. H. B. Garrett, A. Chutjian and S. Gabriel, "Space Vehicle Glow and Its Impact on Spacecraft Systems," J. Spacecraft and Rockets *25*, 321-340 (1988).

2. G.R. Swenson, S.B. Mende, and K.S. Clifton, "Ram Vehicle Glow Spectrum: Implication of NO2 Recombination Continuum," Geophysical Research Letters *12*, 97-100 (1985).

3. L.M. Branscomb, S.J. Smith, and G. Tisone, "Oxygen Metastable Production Through Photodetachment," J. Chem. Phys. *43*, 2906-7 (1965).

4. A. Fontijn, C.B. Meyer, and H.I. Schiff, "Absolute Quantum Yield Measurements of the NO-O Reaction and its use as a Standard for Chemiluminescent Reactions," J. Chem. Phys. *40*, 64 (1964).

5. D.E. Paulson, W.F. Sheridan and R.E. Huffman, "Thermal and Recombination Emission of NO_2," J. Chem. Phys. *53*, 647 (1970).

6. J.-H Yee and A. Dalgarno, "Radiative Lifetime Analysis of the Shuttle Optical Glow," J. Spacecraft and Rockets *23*, 635-40 (1986).

7. O. H. Crawford, "Oscillations in Photodetachment Cross Sections for Negative Ions in Magnetic Fields," Phys. Rev. A *37*, 2432-40 (1988).

8. H. F. Krause, "A Novel High-Resolution Experimental Approach for Studying Laser Photodetachment in a Magnetic Field," Fourth Int. Symp. Prod. and Neutralization of Negative Ion Beams (Brookhaven, N.Y., 27-31 Oct. 1986); and unpublished results (Oak Ridge Nat. Lab., 1989).

9. G.A. Arnold and D.J. Coleman, "Surface Mediated Radical Recombination Luminescence: O + NO + Ni," J. Chem. Phys. *88*, 7147-56 (1988).

LABORATORY INVESTIGATIONS: LOW EARTH ORBIT ENVIRONMENT CHEMISTRY WITH SPACECRAFT SURFACES

Jon B. Cross
Los Alamos National Laboratory
Chemical and Laser Sciences Division
Los Alamos, NM 87545

ABSTRACT

Long-term space operations that require exposure of material to the low earth orbit (LEO) environment must take into account the effects of this highly oxidative atmosphere on material properties and the possible contamination of the spacecraft surroundings. Ground-based laboratory experiments at Los Alamos using a newly developed hyperthermal atomic oxygen (AO) source have shown that not only are hydrocarbon based materials effected but that inorganic materials such as MoS_2 are also oxidized and that thin (750 Å) protective coatings such as Al_2O_3 can be breached, producing oxidation of the underlying substrate material. Gas-phase reaction products, such as SO_2 from oxidation of MoS_2 and CO and CO_2 from hydrocarbon materials, have been detected and have consequences in terms of spacecraft contamination. Energy loss through gas-surface collisions causing spacecraft drag has been measured for a few select surfaces and has been found to be highly dependent on the surface reactivity.

INTRODUCTION

Interest in long-term (20-year) habitation of low earth orbit (LEO) for military and civilian use is increasing and will in all likelihood continue to increase for the foreseeable future. The Russian space station Mir is functional, and the United States is designing its own station--FREEDOM. Operations in LEO (200-500 km) however must take into consideration the high partial pressure (10^{-6} to 10^{-7} torr) of atomic oxygen (AO) that exists at these altitudes. These partial pressures produce extensive oxidation of materials facing the direction of travel (ram direction of spacecraft). The ram oxidation is most severe not only because of the high flux (10^{15} AO/s-cm^2) but also because of the high collision energy (5 eV) of AO with the ram surfaces. Both the high flux and the high collisional energy are a result of the high orbital velocity (8 km/s) of the spacecraft. Los Alamos is performing ground-based simulations of these conditions using a continuous-wave (cw), laser-sustained discharge source for the production of a 1- to 5-eV beam of AO with a flux of up to 1×10^{17} AO/s-cm^2 ($100 \times$ LEO flux).

ATOMIC OXYGEN FACILITY

The AO source[1-3] (Fig. 1) employs a cw plasma formed by focusing a high-power CO_2 laser beam to produce plasma temperatures of 15 000 to 20 000 K in a rare-gas/oxygen mixture. The cw CO_2 laser (10.6 μm) is used to sustain the spark-initiated plasma in the mixture, which subsequently flows through a 0.3-mm-diam nozzle in an isentropic expansion producing an atomic beam of neutral species. Stagnation pressures of 2 to 8 atm are employed depending on the rare gas, i.e., 2 atm for 50% O_2 in argon and 8 atm for 15% O_2 in helium. The isentropic expansion acts as an atomic accelerator where collisions of the excess lighter gas (helium) with the heavier AO bring the AO velocity up to or near that of the helium.[4] A 2.54-cm focal-length ZnSe lens is used to focus the CO_2 laser beam to a 100-μm spot, which produces power densities of 10^7 W/cm^2 and sustains the plasma at a roughly 50% ionized condition. The lens is moved axially to position the plasma ball in the throat of the water-cooled nozzle. We have obtained continuous operation times of >75 hours with fluences of $>10^{22}$ AO/cm^2, roughly equal to the fluence that the U.S. Space Station will receive at 500 km over its operational lifetime (20-30 years). The source is mounted in a molecular beam apparatus (Fig. 2), where the gas mixture is skimmed after exiting the nozzle and then collimated into a neutral atomic beam of rare gas and AO. The facility consists of

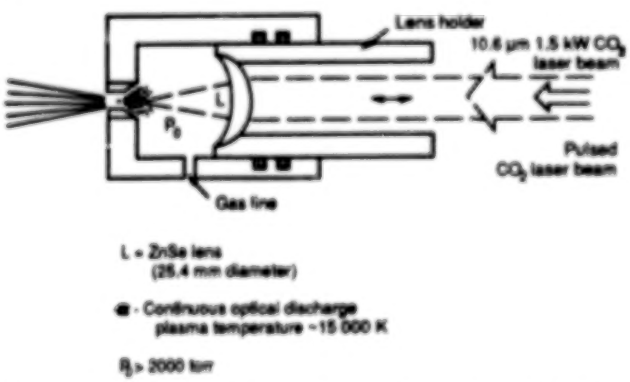


Figure 1. Laser-sustained plasma atomic oxygen beam source. Plasma is initiated by one pulse from a pulsed CO_2 laser and sustained by a 1.5 kW cw CO_2 laser.

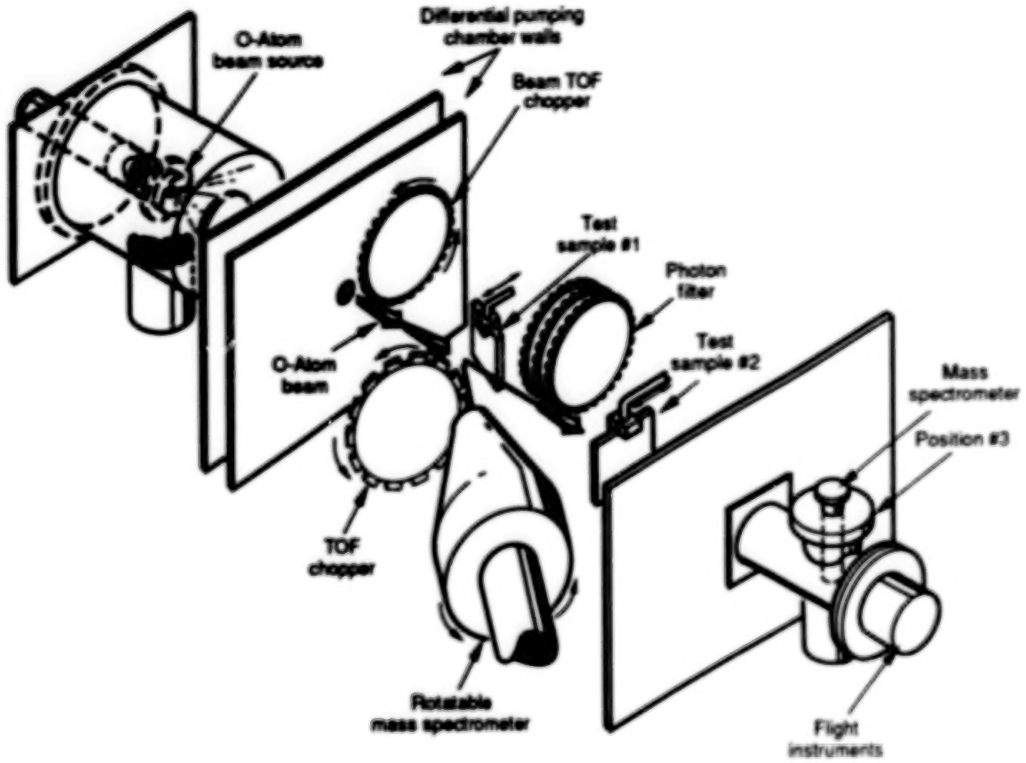


Figure 2. Primary components of the molecular beam apparatus that contains the AO source. The TOF chopper and mass spectrometer at the flight instrument position are used to obtain beam velocity distributions while the rotatable quadrupole mass filter is used to measure angular scattering distributions. The photon filter will be used to filter out plasma produced light from the AO beam to isolate the effects of photons on gas-surface reactions.

- the laser-sustained AO beam source,
- three stages of differential pumping between the source and three sample manipulators located 15, 55, and 120 cm from the source,
- a rotatable mass spectrometer with time-of-flight (TOF) capability for measuring scattered particle angular and velocity distributions to determine drag coefficients and gas-phase reaction products,
- a quadrupole mass spectrometer (located 120 cm in line with the source) used for beam TOF measurements and a mass spectrometer calibration chamber, and
- a photon filter used to filter out photons created in the AO source for studying the effects of vacuum ultraviolet (vuv) radiation on polymer-coating reactions.

At the sample manipulator position, 15 cm from the source, AO flux densities of $\sim 10^{17}$ AO/s-cm^2 are obtained, whereas at the mass spectrometer calibration chamber position, flux densities of $\sim 10^{15}$ AO/s-cm^2 are recorded. A base pressure of 1×10^{-6} torr, which rises to 2×10^{-4} torr when the AO beam is operating, is created in the sample exposure chamber. Figure 3 shows AO energy distributions obtained from TOF analysis of the AO beam. The rare-gas symbols indicate the oxygen/rare-gas composition, i.e., neon designates a mixture of 50% neon + 50% O_2, whereas Ne/He designates 25% O_2 + 25% neon + 50% helium. For comparison, the calculated in-orbit AO energy distribution, assuming an atmospheric temperature of 1000 K, is shown. The O_2 dissociation for the argon, neon, Ne/He, and helium mixtures is 85%, 87%, 96%, and an estimated 98% at stagnation pressures of 1900, 2200, 4200, and 6000 torr, respectively.

Vacuum ultraviolet intensities comparable to that of solar radiation are observed with a Ne/O_2 mixture, whereas Ar/O_2 produces vuv intensities that are a factor of 3 higher; thus, polymer materials can be investigated under the same vuv conditions that exist in orbit. However, the high transmission velocity filter can be used to eliminate photons from the AO beam for investigation of the mechanisms of vuv/AO polymer degradation.

BEAM FLUX CALIBRATION

Samples are normally exposed to the beam at the sample manipulator 15 cm from the nozzle (position #1), and beam calibration is performed at the mass spectrometer calibration position 120 cm from the nozzle (position #3). A simple technique[3] has been developed to perform beam calibrations and to evaluate reaction rates of coatings and conductive materials. A thin strip of conductive material is deposited on a nonconducting and nonreactive substrate, and a resistance measurement is made as a function of time of exposure to the AO beam to determine the rate of material loss or the conversion to a nonconducting oxide. In the case of silver, the conducting metal is transformed into its nonconducting oxide, thus increasing the device's resistance, whereas carbon, if

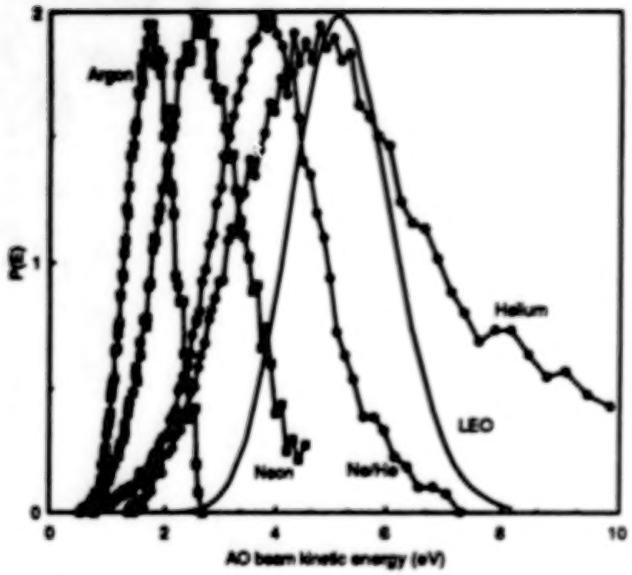


Figure 3. Atomic oxygen kinetic energy distributions. Solid line without data points is approximate orbital kinetic energy distribution of AO with surfaces facing the direction of travel (ram surface).

deposited in place of the silver, forms volatile oxides, which alter the film's cross-sectional area. The time history of a 250-Å-thick bare silver film, when exposed to the AO beam (2.5 eV) at position #3, is shown in Fig. 4 and represents an oxidation rate of roughly 1.7 monolayers/s or a beam flux of approximately 2×10^{15} AO/s-cm^2. In addition, the reactivity of coating[4] can be determined by measuring the time required to burn through a known thickness of the coating. The burn-through is detected by the subsequent oxidation of the silver.

RESULTS

Surface Temperature/Beam Energy Effects

Investigations were performed to determine the relative importance of surface temperature (activation energy) and beam translational energy on the reaction rate of AO with sputter-deposited graphite. Figure 4 shows a typical data set, where a silver strip is used to obtain a beam flux calibration at position #3, and then a sputter-deposited graphite strip is exposed at position #1. The 250-Å bare silver strip required over 100 s before the silver began to react. This effect is caused by the burning off of a hydrocarbon overlayer, which was produced in the device fabrication. A large number of bare silver actinometers have been exposed, and in all cases, the lag or burn-off time correlates with the silver oxidation rate or the AO flux.

Figure 5 shows the reaction efficiency variation of the sputter-deposited graphite with AO over the kinetic energy range of 1.5 to 3 eV at a surface temperature of 35°C. The data represent a 10% to 30% change in reaction efficiency between 1.5 and 3 eV. Similar investigations with varying surface temperatures at a constant beam energy show an activation energy of 800 cal/mole for graphite oxidation, which agrees with shuttle-based flight experiments.[7] These results indicate that graphite-based materials are affected primarily by high kinetic

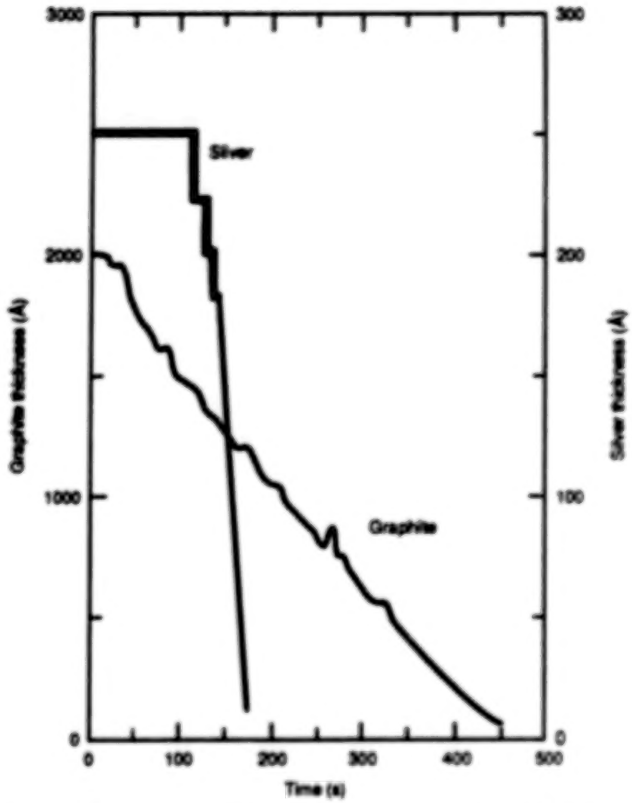


Figure 4. Reaction rates of 250 Å silver film at position #3 and 2000 Å sputter deposited graphite at position #1.

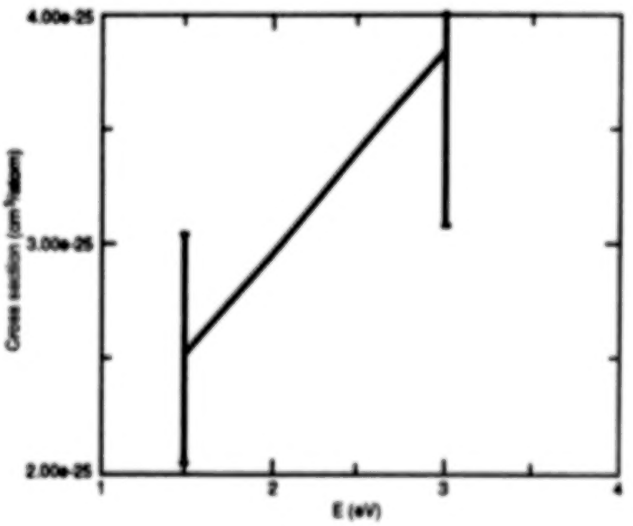


Figure 5. Translational energy dependence of sputter deposited graphite with AO at a surface temperature of 35°C.

AO but that high-temperature surfaces will suffer additional degradation in LEO because of the thermal energy component of the reaction.

Beam Surface Scattering

AO angular scattering distributions from reactive and nonreactive materials were measured at a beam kinetic energy of 1.5 eV using the rotatable mass spectrometer (Fig. 2). These

distributions indicate that the reactive surface produces complete accommodation of the AO and that the AO leaves the surface in a cosine distribution, having forgotten its initial velocity vector (Fig. 6). The nonreactive surface, however, produces specular-like scattering, with the AO remembering its initial velocity vector as it leaves. Separate TOF experiments[1] (not shown here) indicate that approximately 50% of the initial energy is lost to the nonreactive surface, whereas the reactive surface showed complete accommodation. The polyethylene gas-phase products were H_2O, CO, and CO_2, all of which showed accommodation to the material surface temperature (393 K). These results have important implications for long-term operation of spacecraft in the LEO environment in terms of spacecraft drag and contamination. Nonreactive surfaces reduce spacecraft drag more than reactive ones and simultaneously act as mirrors for AO, possibly creating higher than normal AO fluxes on nonram surfaces that are shielded from direct AO attack. The H_2O, CO_2, and CO reaction products are emitted in a cosine distribution from surfaces and may contaminate other parts of the craft. Information on detailed angular/velocity scattering distributions and reaction product identity will allow complete computer simulation of spacecraft interaction with the LEO environment. Predictions can then be made on long-term drag, contamination, and materials degradation, all of which determine the spacecraft operational lifetime.

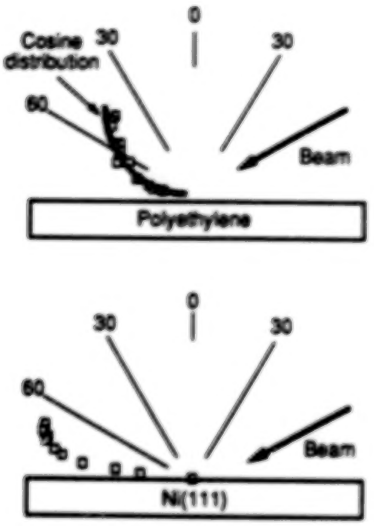


Figure 6. Angular scattering distributions of AO from a reactive surface (cosine) and a nonreactive surface of nickel oxide (specular like).

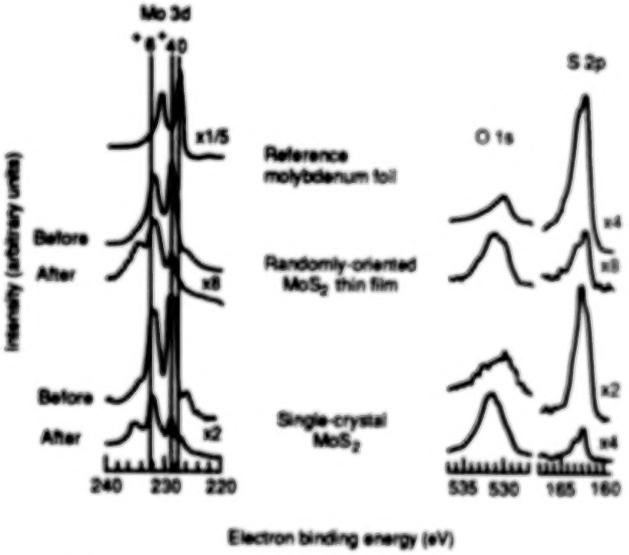


Figure 7. X-Ray photoelectron spectra of reference molybdenum foil, randomly oriented MoS_2 thin film, and single-crystal MoS_2. Note the presence of Mo^{+6} state indicating MoO_3 formation.

Tribology Investigations

In our first study,[8] the oxidation properties of various crystalline forms of MoS_2 were investigated to determine possible tribological implications for lubricating films used on spacecraft in LEO. The following conclusions were drawn from that work:

- Exposure of MoS_2 to energetic (1.5 eV) AO under anhydrous conditions results predominantly in the formation of MoO_3 in the near-surface region with lesser amounts of MoO_2 (Fig. 7).
- The oxide layer is roughly 10-30 monolayers in thickness and consists of a mixture of the oxides with MoS_2.
- The extent of oxidation is essentially independent of crystallographic orientation of the MoS_2.
- Diffusion of oxygen atoms through the oxide layer is very slow.

The reaction independence from crystallographic orientation is not surprising since exposure to AO bypasses the dissociative adsorption step in O_2 oxidation whose rate is highly dependent upon surface defects and crystallographic orientation.

The next study[9] showed that SO_2 is generated and outgases from the MoS_2 surface (Fig. 8) with a reactivity comparable to that of kapton, and the reaction of AO with MoS_2 has little or no translational energy barrier, i.e., thermally generated AO reacts as readily (within factors of 2-4) as that having 1.5 eV of translational energy. It was discovered that if water is present, sulfates are formed on the MoS_2 surface and that the sulfate is most likely in the form of sulfuric acid. These results imply that water dumps or outgasing in LEO have the potential to form sulfuric-acid-covered surfaces on MoS_2 lubricants.

Friction studies were undertaken, and it was found that a 1-micron-thick sputter-deposited MoS_2 film that was exposed to the hyperthermal AO beam produced a very high (0.2) friction coefficient because of oxide formation on the surface. The friction dropped very quickly as the 100 Å oxide film was worn away exposing fresh MoS_2 material (friction coefficient = 0.05). The friction measurements indicate that a high initial starting force would be required after MoS_2 exposure to AO with the force dropping to its normal low value after several cycles of operation. A continuous flux of AO striking the MoS_2 surface would result in a higher average value of the friction depending upon the speed per cycle and absolute value of the AO flux.

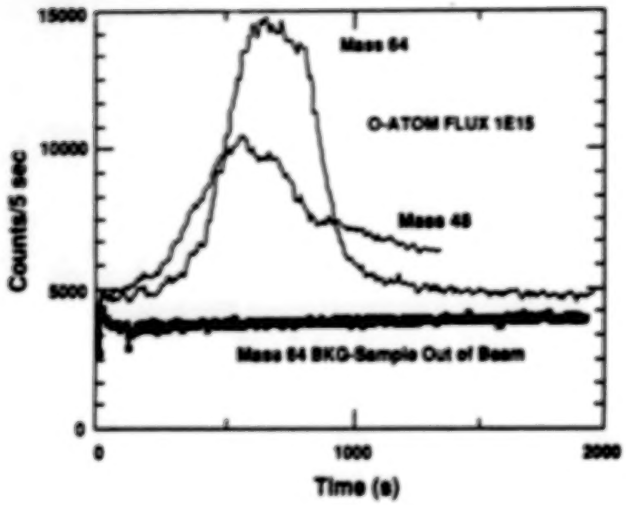


Figure 8. The mass spectrometric signal showing the outgassing rate of SO_2 and SO from a MoS_2 surface exposed to the indicated AO flux.

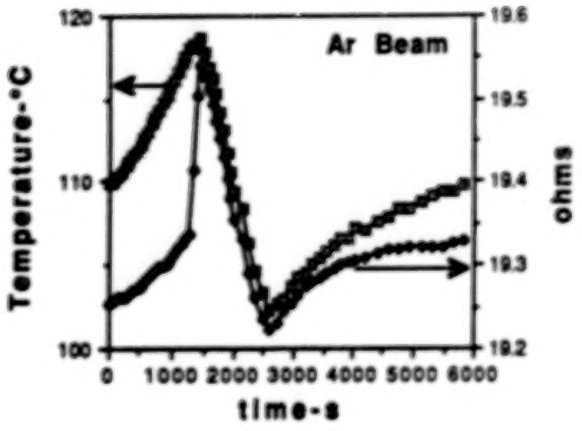


Figure 9. Results of heating the Al_2O_3-coated silver strip during exposure to a 3-eV neutral argon beam.

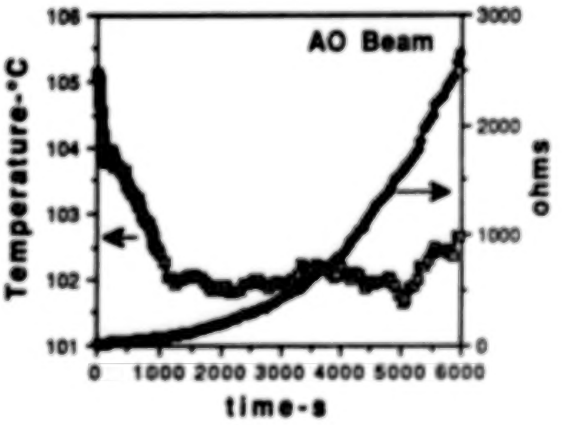


Figure 10. Results of heating the coated silver during exposure to a 1.5-eV AO beam.

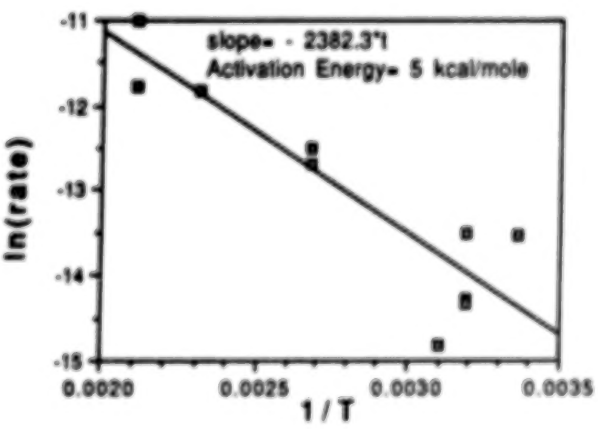


Figure 11. Summary of temperature dvariation of the coated silver reaction rate.

Protective Coatings

Because protective coatings will be needed to impede the degradation of specialized materials for solar arrays, solar concentrators, and mirrors, etc., a series of studies on the durability of thin inorganic coatings has been undertaken. Thin films (250 Å) of silver were overcoated with sputter-deposited Al_2O_3 (750-Å thickness) on rough (1-micron rms roughness) alumina and also on smooth single crystal sapphire substrates. Resistance measurements were taken on the coated silver as a function of substrate temperature while being exposed to the AO beam. The reaction rates measured on the coated silver on the rough alumina substrate were not reproducible presumably because of the inability to uniformly coat over the high roughness of the substrate. The reaction rates measured on the smooth sapphire substrates though showed a much greater reproducibility than those measured on the alumina. Figures 9 and 10 show some typical data obtained using the sapphire substrate. Figure 9 shows the results of heating the Al_2O_3-coated silver strip during exposure to a 3-eV neutral argon beam having a flux of $\approx 10^{17}$ Ar/s-cm^2. The resistance changes correlate with the substrate temperature, indicating that the silver is not oxidized by reaction with the substrate or coating but rather that the resistance is only changing because of the temperature coefficient of resistance. Note that the absolute resistance change is only on the order of 0.5 ohms. Figure 10 on the other hand shows the results of heating the coated silver during exposure to a 1.5-eV AO beam having a flux of $\approx 10^{17}$ AO/s-cm^2. The resistance change does not correlate with small temperature variations, and the absolute resistance change is on the order of 3 K ohms, indicating that the presence of hyperthermal AO causes sever degradation of coated silver. Figure 11 summarizes the temperature variation of the coated silver reaction rate, i.e., the activation energy is ≈ 5 kcal/mole. This activation energy is an order of magnitude lower than the diffusion-limited activation energy for high-temperature O_2 interaction with Al_2O_3. Further experiments are under way to understand the degradation mechanism, i.e., whether there are microscopic flaws in the film that

allow AO to diffuse along grain boundaries and dislocations to the silver or whether high translational energy AO produces enhanced bulk diffusion in Al_2O_3, etc. Nevertheless, these initial experiments have shed light on the difficulty of producing protective coatings that will withstand long-term exposure to this highly unusual environment.

Accelerated Testing

Spacecraft operating in LEO for extended periods of time (20 years) will experience total fluences of 10^{22} to 10^{23} AO/cm^2 (Ref. 10). Materials certification studies are needed to ensure reliable operation of a spacecraft over its operational lifetime and thus will require exposure of selected materials to the same total fluence that the craft will experience but in less than real time. These studies can be performed with the laser-sustained AO plasma source because of the ability to obtain a total fluence of 3.6×10^{22} AO/cm^2 at the 1-5 eV energy level in 100 hours of operation. In addition to the ability to perform such exposures, an understanding of the surface reaction mechanisms is needed to interpret the results. For example, the increased AO flux needed for accelerated testing may provide a mechanism for enhanced AO recombination on surfaces before reaction with the surface can occur, thus giving a lower materials reaction probability than actually occurs in orbit. The mechanistic studies needed to interpret accelerated testing data are now under way.

Orbital Flight Experiments

The LEO simulation performance requires comparison of ground-based results to the results produced by the actual LEO environment in order to have complete confidence in the simulation fidelity. We are participating in two flight experiments which will provide data that can be directly correlated to our ground-based results. We provided upwards of 50 conductive strip[9] samples that have been placed on a small satellite for exposure to the in-orbit LEO environment. Real-time monitoring will be performed through telemetry of the conductive strip resistances as a function of exposure time. Similar experiments with identical samples will be performed using the AO plasma source and will be compared to the flight results. This comparison will provide a good test of simulation fidelity. In addition, we are participating in a Shuttle-based experiment that is being run from NASA/Johnson Space Center (EOIM-3) in which a mass spectrometer will take measurements of the LEO ambient AO flux as well as observe the gas-phase reaction products produced by the LEO AO interaction with various samples. The flight mass spectrometer (AFGL) is being calibrated in our beam facility in order to fully compare the in-orbit data with our simulation results.

CONCLUSION

The design of reliable, long-term low earth orbit in space operations, automation, and robotic systems will require a broad knowledge base in "chemistry under extreme conditions." The few examples detailed in this paper reinforce the idea that the chemistry under LEO conditions is very different from that experienced under geosynchronous orbit or deep-space environments and that equipment designed for those conditions will, in general, need to be modified or protected for use in LEO.

ACKNOWLEDGMENTS

The author thanks Frank Archuleta for his expert craftsmanship, dedication, and hard work, all of which made this project successful. The author wishes to acknowledge discussions with Lubert Leger, James T. Visentine, and Steve Koontz of NASA Johnson Space Center. We also wish to thank Steve Koontz for scanning electron microscopy (SEM) and profilometer analysis of the kapton and Teflon samples and to acknowledge funding for this work from NASA Johnson Space Center. The author wishes to acknowledge funding from McDonnell Douglas Corporation and collaboration of Esther Lan and Charles Smith in the development of the thin strip conductive material technique for ground-based simulation/LEO comparison. The author wishes to acknowledge funding from the Air Force Geophysics Laboratory and the collaboration of Don Hunton, John Ballenthin, and Lt. Ross McNutt of AFGL in developing mass spectrometer calibration techniques. The Strategic Defense Initiatives Office is thanked for the opportunity to participate in a satellite flight experiment to compare ground-based/LEO exposure results with orbital environment exposure experiments.

REFERENCES

1. Cross, J. B., Cremers, D. A., Spangler, L. H., Hoffbauer, M. A., and F. A. Archuleta, "CO_2 Laser Sustained CW Discharge Atomic Beam Source," in *Proceedings of the 15th International Symposium on Rarefied Gas Dynamics*, Vinicio Boffi and Carlo Cercignani, Eds. (Tuebner, Stottgart, 1986), Vol. 1, pp. 657-666.

2. Cross, J. B. and Cremers, D. A., "High Kinetic Energy (1-10 eV) Laser Sustained Neutral Atom Beam Source," *Nucl. Instru. & Methods* **B13**, 1986, 658.

3. Cross, J. B. and Blais, N. C., "High Energy/Intensity Atomic Oxygen Beam Source," 16th International Symposium on Rarefied Gas Dynamics, Pasadena, California, July 11-15, 1988, Los Alamos National Laboratory document LA-UR-88-2188 (proceedings in press).

4. *Methods of Experimental Physics* (Atomic Interactions), Benjamin Bederson and Wade L. Fite, Eds. (Academic Press, New York and London, 1968), Vol. 7, Part A.

5. Cross, J. B., Lan, E. H., and Smith, C. A., "A Technique to Evaluate Coatings for Atomic Oxygen Resistance," in *33rd International SAMPE Symposium and Exhibition* **33**, 1988, 693.

6. Cross, J. B., Lan, E. H., and Smith, C. A., "Evaluation of Coatings for Atomic Oxygen Protection in Low Earth Orbit," Fourth International Symposium on Spacecraft Materials in the Space Environment, Toulouse, France, September 6-9, 1988, Los Alamos National Laboratory document LA-UR-88-2565.

7. Gregory, J. C., "The Reaction of 5 eV Oxygen Atoms with Polymeric and Carbon Surfaces in Earth Orbit," *Polymer Preprints* **28**, 1987, 459.

8. Martin, J. A., Cross, J. B., and Pope, L. E., "MoS_2 Interactions with 1.5 eV Atomic Oxygen," Materials Research Society, 1988 Fall Meeting, Boston, Massachusetts, November 28-December 3, 1988, Los Alamos National Laboratory document LA-UR-88-1977.

9. Cross, J. B., Martin, J. A., Pope, L. E., and Koontz, S. L., "Oxidation of MoS_2 by Thermal and Hyperthermal Atomic Oxygen," Third International SAMPE Electronics Materials and Processes Conference, Los Angeles, California, June 20-22, 1989, Los Alamos National Laboratory document LA-UR-89-956.

10. L. Leger, J. Visentine, and B. Santos-Mason, "Selected Materials Issues Associated with Space Station," *SAMPE Quarterly* **18**, 1987, 48.

ATOMIC OXYGEN EFFECTS ON SPACECRAFT SURFACES: ORBITAL MEASUREMENTS AND GROUND SIMULATION

J. Gregory
University of Alabama

(Paper not provided by publication date.)

RTDS: REAL-TIME EXPERT SYSTEMS IN SHUTTLE MISSION CONTROL

J. Muratore, A. Rasmussen, and M. Gnabasik
MITRE Corporation

J. Dingler
NASA/Johnson Space Center

(Paper not provided by publication date.)

Artificial Intelligence for Multi-Mission Planetary Operations

David J. Atkinson Denise L. Lawson Mark L. James

Jet Propulsion Laboratory/California Institute of Technology
M.S. 301-440
4800 Oak Grove Drive
Pasadena, California 91109
USA

ABSTRACT

This paper gives a brief introduction to an automated system called the "Spacecraft Health Automated Reasoning Prototype" (SHARP). SHARP is designed to demonstrate automated health and status analysis for multi-mission spacecraft and ground data systems operations. The SHARP system combines conventional computer science methodologies with artificial intelligence techniques to produce an effective method for detecting and analyzing potential spacecraft and ground systems problems. The system performs real-time analysis of spacecraft and other related telemetry, and is also capable of examining data in historical context. Telecommunications link analysis of the Voyager II spacecraft is the initial focus for evaluation of the prototype in a real-time operations setting during the Voyager spacecraft encounter with Neptune in August, 1989. This paper will report on the preliminary results of the SHARP project and discuss plans for future application of the technology.

INTRODUCTION

The Voyager 1 and Voyager 2 spacecraft were launched from Cape Canaveral, Florida, on August 20, 1977. The technology to track and monitor such probes was designed and developed in the early 1970's. During critical periods of the mission, up to 40 real-time operators are required to monitor the spacecraft's 10 subsystems on a 24-hour, 7-day-per-week schedule. This does not include the numerous subsystem and scientific instrument specialists who must constantly be available on call to handle emergencies. To accomodate the increasing load on mission operations in the 1990's when the Voyager, Magellan, Galileo, Ulysses, Mars Observer, CRAF, and Cassini spacecraft are all flying, JPL has initiated an effort to coordinate all missions through one multi-mission team in the Space Flight Operations Center (SFOC) that operates all spacecraft.

The Spacecraft Health Automated Reasoning Prototype (SHARP) is an on-going effort to apply artificial intelligence (AI) techniques to the task of multi-mission monitoring and diagnosis of spacecraft and ground systems, and to demonstrate those capabilities in tough, operational settings to prove their performance. The Voyager 2 spacecraft and its Telecommunications subsystem were targeted for the initial SHARP demonstration. The spacecraft's August 1989 encounter with the planet Neptune afforded an excellent opportunity to evaluate SHARP in an intense operational environment. The Telecommunications subsystem suffers from frequent anomalies and also requires coordination of monitoring and diagnosis efforts of both the spacecraft and ground data systems (GDS). Finally, due to cumbersome and time-consuming manual processes and obsolete technology, severe limitations exist on the current method of analyzing Voyager telecommunications data. Quite simply, this was both an operations area sorely in need of automation as well as one of the most challenging to automate.

DESCRIPTION OF SHARP

SHARP introduces automation technologies to the spacecraft monitoring process to eliminate much of the tedious analysis associated with analyzing and responding to spacecraft and ground system anomalies. SHARP also automates many of the more mundane, manual processing activities in mission operations. The SHARP system features on-line data acquisition of all required information for monitoring the spacecraft and diagnosing anomalies. The data is centralized into one workstation, which serves as a single access point for the aforementioned data as well as for the diagnostic heuristics. Figure 1 illustrates a

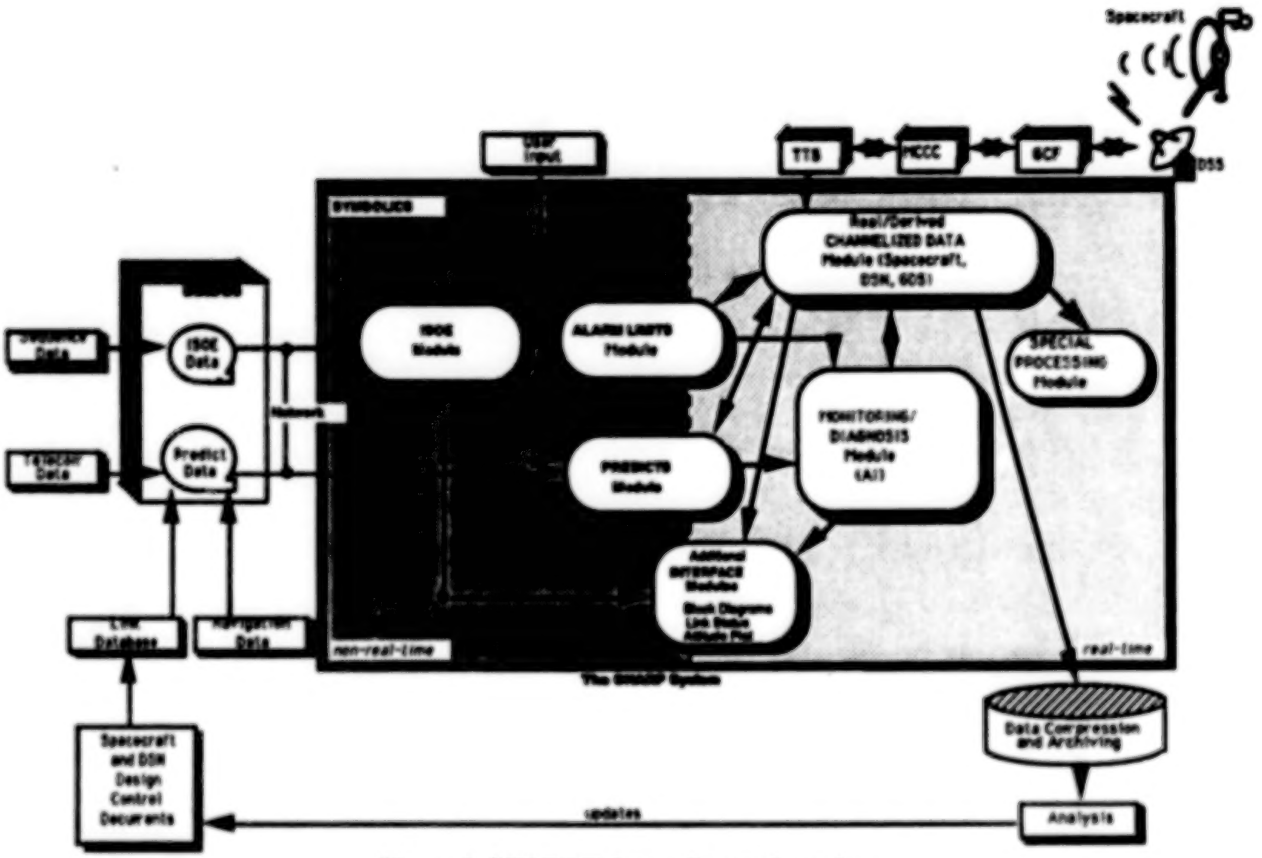


Figure 1: SHARP Telecom System Overview

top-level view of the SHARP system. Shown are the individual modules that comprise the system, as well as relevant components that are external to the Voyager application of SHARP. This section describes these components and related SHARP features in more detail.

The SHARP system provides numerous sophisticated graphical displays for spacecraft and station monitoring. A comprehensive user interface has been developed to facilitate rapid, easy access to all pertinent data and analysis. An interface exists for each major module of the SHARP system. Each interface provides customized functions that allow data specific to that module to be easily accessed, viewed, and manipulated. Each SHARP module can be accessed from any other module at any time, and all displays are in color with mouse sensitivity and menu-driven commands.

SHARP is implemented in Common LISP on a SYMBOLICS 3650 color LISP machine. Many components of the system utilize STAR*TOOL [1] (patent pending), a language and environment developed at JPL which provides a toolbox of state-of-the-art techniques commonly required for building AI systems. The SHARP system is a moderately large system, consisting of approximately 354,000 lines of Common LISP source code, including code specifically developed for SHARP, knowledge bases, STAR*TOOL functions, and Symbolics Genera™ supporting code.

Predicts

To later evaluate spacecraft performance, predictions, or "predicts," of a set of critical parameters are usually obtained in advance for particular spacecraft subsystems. Numerical simulations are frequently employed in the generation of predicts. The SHARP system captures raw telecommunications predicts which are output from an existing computer program. Pass predicts are automatically generated at 15-second intervals, the shortest possible time interval between the arrival of any two spacecraft data points. Instantaneous predicts, which are pass predicts corrected in real-time for spacecraft pointing loss and Deep Space Station (DSS) system noise temperature, are also automatically calculated at 15-second intervals. Spacecraft and DSS residuals, difference measurements between the actual values and predicted values, are automatically derived in real-time. These functions alone can save up to two hours of operator time each shift.

The Predicts interface in SHARP allows tabular display of raw predicts, pass predicts, instantaneous predicts, and re-

siduals for any specified time range. A color-coded DSS availability graph has also been provided which enables rapid identification of available stations for any given viewing period. Situations that mandate that another Deep Space Station be acquired can be addressed immediately as opposed to the more arduous current method, which requires the manual look-up of each station at the specified time period.

Integrated Sequence of Events

The SHARP system also acquires the Integrated Sequence of Events (ISOE), from an external computer. The system provides an ISOE interface which offers numerous capabilities to the operator. A generic capability to extract subsystem-specific information from the ISOE has been developed, e.g., Telecom-specific events may be stripped from the ISOE and displayed to enable rapid identification of significant Telecom activities to be monitored during any particular pass. Editing of the ISOE, e.g., to reflect real-time commanding of the spacecraft, may be performed with ease. This reduces the likelihood of referencing outdated material. Editing is accomplished via menu-driven commands that contain explanations of the complex ISOE data. For example, CC3A32330 means that the x-band modulation index is 32, the two drivers are on, the subcarrier frequency is high, and the data line rate is high. Translation of these spacecraft commands from their raw form into more understandable summaries of spacecraft activity may be performed, and the user can request status summaries of any activity. A history display is maintained as the ISOE is updated so that the user can verify modifications.

Channelized Data Plotting

SHARP includes very flexible plotting capabilities for channelized data. Using simple menu-driven commands and program parameters, an operator can construct an arbitrary number of data plots on an arbitary number of screens, a significant improvement over existing capabilities. The user dynamically customizes the display at any time by selecting which and how many channels to view, the time scale, the data range for each plot, and even the icon to use for graphing points on each channel. Each plot is color-coded by the user for easy visual distinction between displayed channels. When any channel is in alarm, its corresponding data points are plotted in red, facilitating rapid detection of an alarm condition. The channel's associated alarm limits may be optionally overlaid onto the channel's plot for further information. Each data point is mouse-sensitive to provide time and numerical value indicators, and an automatic counter continually indicates the number of data points per plot. Pan and zoom features augment this display, which can represent information as graphs of actual or derived data vs. time, xy plots, scatter plots, or logarithmic scales.

Dynamic Alarm Limits

SHARP automatically determines alarm limits in real-time. The selection of alarm limits by SHARP accurately reflects each spacecraft or DSS configuration change as a result of an automated analysis of the ISOE, Predicts, and the real-time channelized engineering data from the spacecraft. Dynamic alarm limit determination eliminates the cumbersome alarm change paperwork process as well as many occurrences of false alarms.

Alarm tables, representing spacecraft configurations, are also available to operators on-line within the SHARP system. SHARP provides a user interface which allows viewing and editing of established spacecraft engineering alarm limits, DSS performance limits, ground data system limits, and residual thresholds. Authorized users may permanently alter any of these limits, and specified values may be changed temporarily for the remainder of that particular spacecraft pass. The latter capability, manual override, enables alarm suppression or closer scrutiny for any particular event, with no intervening paperwork.

System Status Displays

SHARP provides several graphical displays which represent system-wide status, including system configurations, and on-going or predicted events. SHARP automatically highlights alarmed events as they occur, and indicates the location of problems as well as the probable causes of the alarms. These displays are possible in SHARP because of the centralization of a wide variety of data sources in the system as well as its highly integrated architecture.

One such display shows a comprehensive view of Telecom link status over a settable, multiple hour time period. The display is continuously updated in real-time. Actual station coverage is illustrated, along with spacecraft transmitter power status, data rate, data outages, and real-time recording of station uplink (signal transmission) and projected downlink a round-trip light time later. Detailed analysis is performed and information is subsequently color-coded to represent changes in status. The display provides the user with such valuable information as time ranges and explanations of data outages (e.g., no station coverage or ongoing spacecraft maneuver), and can warn the operator when to expect noisy data and why.

Telecommunications system status may also be monitored using displays of functional block diagram schematics. Schematic diagrams are provided for the end-to-end communications path from the spacecraft through the Deep Space Communications Complex and Ground Communications Facility (GCF) to the Mission Control and Computing Center (MCCC) at JPL and final destina-

tion of the Test and Telemetry System (TTS) computers. The diagrams are organized hierarchically, with increasingly detailed schematics at the lowest levels. An operator can shift among the diagram displays with mouse-clicks on elements of the diagram or simple menu commands.

The appearance of the schematic diagrams is dynamically driven by changes in the ISOE, channelized data, alarm conditions, and diagnostic conclusions. The status of spacecraft and DSS components (operational, off-line, or in alarm) is depicted by color, facilitating rapid status identification at a glance.

Special Analysis Modules

The SHARP system also contains special processing modules to perform subsystem-specific analyses. Such modules are easily integrated with SHARP and can make use of the system's alarming, plotting, and diagnostic capabilities. For the Telecommunications subsystem, a special processing module which performs a Fast Fourier Transform (FFT) on DSS receiver automatic gain control (AGC) data and analyzes the results was implemented. SHARP analyzes the conical scanning component of the FFT to determine whether the DSS antenna is going off point. This is a relatively common event, which currently may take hours to detect and correct. Spacecraft and scientific information can be permanently lost when this situation occurs. SHARP's FFT display illustrates the results of a FFT process performed on 64 data points of a particular channel, and provides instant information on conical scan error. The problem can be detected in a matter of minutes, and the station can be contacted to correct the antenna movement prior to the loss of contact with the spacecraft.

Artificial Intelligence

Many of the modules of SHARP which are based on artificial intelligence techniques are written in a knowledge-based system development language and environment called STAR*TOOL. STAR*TOOL is a Common LISP-based programming language designed at JPL to achieve high computational performance in a suite of tools and techniques commonly required by research, development, and delivery of knowledge-based systems. High computational performance is essential in the scale-up of laboratory pilot systems to operational artificial intelligence systems. Since real-time performance was a baseline requirement for SHARP, STAR*TOOL became one of the fundamental tools used in system development.

AI techniques are distributed throughout all components of the SHARP system. Intelligent programming methodologies such as heuristic adaptive parsing, truth maintenance, and expert system technology enable more effective automation and thorough analysis for most SHARP functions. Artificial intelligence techniques used throughout SHARP, even for "conventional" processes such as alarm limit selection or database maintenance, have proven to be essential to real-time fault detection and diagnosis as well as the high degree of accuracy and precision in these modules. While fault detection, diagnosis, and recovery recommendation functions are localized in the "AI Module" (the structure of which is illustrated in Figure 2 and discussed below), our experience in SHARP development is that artificial intelligence functions should not be simply "layered" on a system, but must be designed as an integral aspect of the complete application.

A blackboard architecture, provided by STAR*TOOL, serves as a uniform framework for communication within the heterogeneous multi-process environment in which SHARP operates. Generally, when two or more processes are cooperating, they must interact in a manner more complicated than simply setting global variables and passing information along such paths. The STAR*TOOL blackboard message system in SHARP provides a standardized method of communication and shared-control between multiple processes for data which can not be efficiently shared through SHARP's centralized database. SHARP's individual diagnostic procedures utilize the blackboard for requesting, retrieving, and combining dependent or ancillary diagnoses.

Heuristic adaptive parsing, a technique from natural language processing, is utilized in the analysis of the raw predicts database which SHARP takes as input. Periodically the format of this data source changes without notification. This has little effect on the human mission operators who read the raw data in tabular form, but for the automated SHARP system, it would require the raw predicts parser to be rewritten to incorporate the new format. To solve the problem, SHARP utilizes Augmented Transition Network (ATN) [2] techniques to accomplish adaptive parsing. The advantage of such an ATN lies in its ability to parse the database according to semantic content rather than syntactic structure. The raw predicts database can therefore be modified and yet remain successfully parsable by SHARP without programmer intervention. This heuristically controlled, format-insensitive parsing ensures continuity despite format modifications in the generation of the raw predicts database.

The centralized database of the SHARP system serves as a central repository of all real-time and non-real-time data, and functions as a local buffer to enable rapid data access for real-time processing. Numerous database manipulation functions have been implemented, and database daemons have been constructed to implement spon-

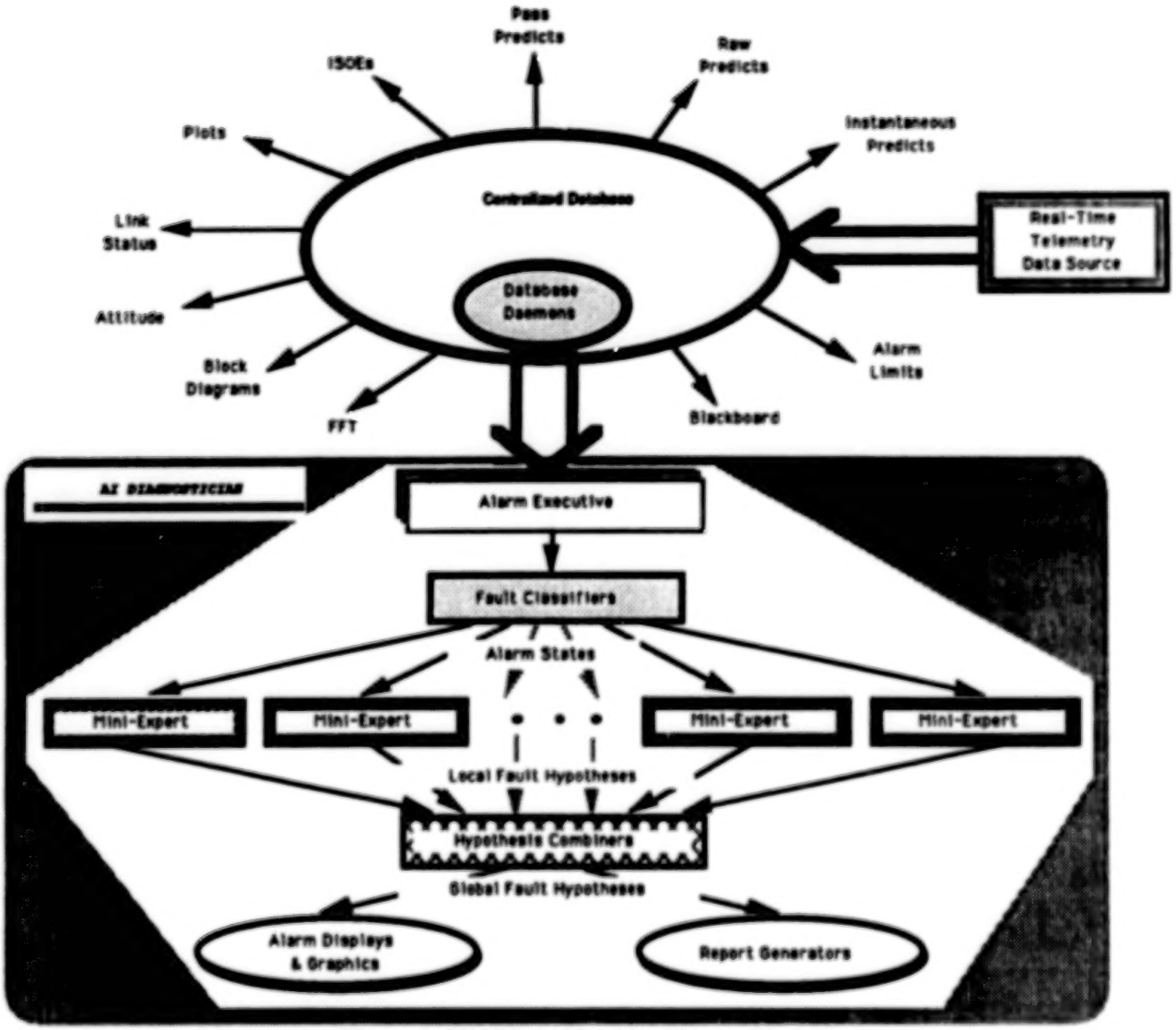


Figure 2: SHARP Artificial Intelligence Module

taneous computations [3]. Requests can be made to the database to trigger arbitrary computations when a complex combination of past, present, and future events occur. A wide selection of retrieval methods by time or value highlight the flexibility inherent in the database. Requests to the database can be made from both AI and non-AI modules of SHARP, and can be handled serially or in parallel.

Various SHARP modules represent and manipulate data symbolically rather than numerically so that particular numeric values can change without forcing the algorithms themselves to be modified. For example, to determine if a channel is in alarm, the rule interpreter manipulates one symbolic fact, "ChannelInAlarm", rather than the many numeric operations that are required to make an actual determination. This is a significant advantage as SHARP presently analyzes over 100 channels, and the alarm determination process varies from channel to channel. Symbolic representation and manipulation of data also simplifies the exchange of information between SHARP modules and reduces reliance on specific dimensionless numeric values. When needed, the specific numeric quantities underlying the symbolic data can be retrieved.

The diagnostic component of SHARP is composed of a hierarchical executive diagnostician coupled with specialized diagnostic systems, called "mini-experts". Each mini-expert is responsible for the local diagnosis of a specific fault or class of faults, such as particular channels in alarm, conical scan errors, or loss of telemetry. The decomposition of diagnostic expertise in SHARP has important and beneficial implications. The localized knowledge in each mini-expert can be maintained and modified inde-

pendently of the other mini-experts. This has allowed efficient, incremental development of SHARP's knowledge bases and should improve the long-term maintainability and extendability of SHARP's knowledge. Localized knowledge also allows each mini-expert to exert more direct control over reasoning processes than would be possible in a "flat" knowledge base with awkward control constructs (a typical failing of pure production rule systems). Efficient control over reasoning processes is required for SHARP to attain real-time performance.

The specialized mini-experts can be either cooperating or non-cooperating. A non-cooperating mini-expert focuses only on its designated fault area, but a cooperating expert has the additional capability of searching beyond its local area to identify related faults that are likely to occur, and in the process thereby invoke other mini-experts to pursue the implications of these hypotheses. Cooperating experts are used in situations where the identification of a particular fault cannot be made by examining a single fault class alone.

The executive diagnostician reviews the overall situation and synthesizes hypotheses which propagate from each mini-expert. This has the effect of simplifying the ultimate diagnosis and recommendations for corrective actions which are presented to the human operator. When multiple, possibly conflicting fault hypotheses are generated, the system lists all possible causes of the anomaly and ranks each according to plausibility. In some cases, SHARP simply has insufficient data to resolve these conflicts. In other cases, conflicts may represent gaps in knowledge which must be resolved by the knowledge engineer through interaction with the domain expert. Bayesian inference processes are used for comparing multiple hypotheses and for prioritizing conflicting fault hypotheses. Bayesian inference procedures also perform uncertainty management to allow continued high performance in the presence of noisy, faulted, or missing data.

If one or more of the cooperating experts fails, the executive diagnostician will continue to operate with only a reduction in the area of local diagnosis that would have been derived from the failed mini-experts. Similarly, if the executive diagnostician fails, the cooperating mini-experts will locally diagnose the faults in isolation of multiple fault consideration, i.e., they degrade into non-cooperating mini-experts.

The executive diagnostician and mini-experts are implemented using rules that execute in pseudo-parallel pursuit of multiple hypotheses. Pseudo-parallelism (i.e., software-based, as opposed to true hardware parallelism) is implemented in SHARP using facilities provided by STAR*TOOL, which includes parallelism as a fundamental control structure. The various diagnostic rules operate in isolation of one another by executing in independent contexts [4] provided by the STAR*TOOL memory model, and communicate through the blackboard facility as described earlier. Contexts can be organized into a tree-like structure to represent contradictory information resulting from changes in facts or from the introduction of new or contradictory hypotheses.

Facilities in the STAR*TOOL truth maintenance system [5] used in SHARP handle data- and demand-driven diagnoses to ensure an appropriate balance between the persistence of hypotheses and sensitivity to new data. The truth maintenance system constantly monitors for violations of logical consistency. For example, it performs conflict checking to maintain consistency among multiple rule firings, hypotheses, and the knowledge base, and allows the context-sensitive management of alarms through a complex response system to combinations of alarm conditions. Truth maintenance techniques also provide a variety of functions for temporal reasoning in multiple fault diagnosis.

CONCLUSIONS

Spacecraft and ground data systems operations present a rigorous environment in the area of monitoring and anomaly detection and diagnosis. With a number of planetary missions scheduled for the near future, the effort to staff and support these operations will present significant challenges.

The SHARP system is an attempt to address the challenges of a multi-mission monitoring and troubleshooting environment by augmenting conventional automation technologies with state-of-the-art artificial intelligence. The artificial intelligence technology used in SHARP will endow mission operations with considerable benefits. Results of this effort to date have already begun to show significant improvements over current Voyager methodologies and have demonstrated potential enhancements to several aspects of Voyager operations. In as many areas as are automated, expert knowledge will be captured and permanently recorded, reducing the frenzied state that occurs when domain specialists announce their impending retirement. Cost reductions will occur as a result of automation and decreased requirement for 24-hour real-time operator coverage. Telecommunications domain experts have said that an application of SHARP could allow as much as a factor of five reduction in the real-time Telecom operations workforce, with comparable savings in other real-time operations areas. Automated fault detection and analysis, which present results to human operations in seconds, have greater accuracy than human manual analyses and will facilitate quicker response to mis-

sion anomalies. The time savings afforded by SHARP-like capabilities, especially during periods of unmanned operation or during emergencies, could mean the difference between the loss or retention of critical data, or possibly even of the spacecraft itself.

ACKNOWLEDGEMENTS

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The authors wish to acknowledge the following people for their participation in the SHARP project: Harry J. Porta and Gaius Martin, software development; Boyd Madsen, expert knowledge for Voyager and Galileo spacecraft telecommunications; and Bruce Elgin and Erann Gat, software contributions. An earlier, and expanded, version of the paper was presented at the 1989 Goddard Conference on Space Applications of Artificial Intelligence, Greenbelt, Maryland, May 16-17 and is included in the proceedings of that conference.

REFERENCES

[1] James, Mark, and Atkinson, David, *STAR*TOOL — An Environment and Language for Expert System Implementation*, NTR C-17536, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, August 19, 1988.

[2] Woods, W.A., "Transition Network Grammars for Natural Language Analysis," COMMUNICATIONS OF THE ACM, New York, NY, Vol. 13, No. 10, October, 1970, pp. 591–606.

[3] Reiger, Chuck, "Spontaneous Computation and Its Role in AI Modeling," in PATTERN-DIRECTED INFERENCE SYSTEMS, Ed., Waterman, D.A., and Hayes-Roth, Frederick, Academic Press, Inc., Orlando, Florida, 1978, pp. 69-99.

[4] McDermott, D. V., "Very large PLANNER-type data bases," Memo AIM-339, AI Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1975.

[5] Doyle, Jon, "A truth maintenance system," ARTIFICIAL INTELLIGENCE, Amsterdam, The Netherlands, Vol. 12, No.3, 1979.

CLEAR: Automating Control Centers with Expert System Technology

Peter M. Hughes
Automation Technology Section / Code 522.3
NASA / Goddard Space Flight Center
Greenbelt, Maryland 20771

Abstract

The Communications Link Expert Assistance Resource (CLEAR) is a fault-isolation expert system to be utilized in the operational environment of the Cosmic Background Explorer (COBE) Mission Operations Room (MOR). CLEAR will assist the COBE Flight Operations Team (FOT) during periods of real-time data acquisition by isolating faults in the spacecraft communication link with the Tracking and Data Relay Satellite (TDRS), providing advice on how to correct them, and logging the events for post-pass evaluation.

After a brief introduction to the problem domain, this paper describes the system requirements, tool selection, development approach, system operation and lessons learned during the transformation of the system from the prototype to the delivered, operational system.

Introduction

The Cosmic Background Explorer (COBE) is a scientific satellite that will carry three instruments to allow scientists to investigate the possible origins of the universe. This satellite will utilize the Tracking and Data Relay Satellite (TDRS) for four or five 20-minute real-time communication events daily. These events will primarily be used for uplinking stored commands, ranging, and monitoring of the satellite's health and safety.

The Flight Operations Analysts (FOAs) in the Payload Operations Control Center (POCC) are responsible for the health, safety, command and control of the COBE spacecraft. This includes the monitoring of the communications link between the COBE and TDRS which demands the real-time evaluation of more than 100 TDRS and spacecraft performance parameters. In order to isolate problems and to select appropriate courses of action for resolving them, this evaluation of real-time data must be correlated with a comprehensive understanding of the TDRS and COBE systems and their communications services. The task is complex, and, if not handled quickly and properly, can result in poor utilization of TDRS services, inefficient spacecraft operations and a potential hazard to the spacecraft's health and safety.

At present, extensive training and communication of actual experience are used to develop the capabilities of the COBE FOAs. Regardless, the size and complexity of the task places a large burden on the analyst.

The Communications Link Expert Assistance Resource (CLEAR) is a fault-isolation expert system to be utilized in the operational environment of the COBE Mission Operations Room (MOR) (the COBE unique portion of the POCC). This expert system was developed to provide quick problem detection and isolation in the communications link thus producing a more efficient and reliable system of operations.

CLEAR will execute on one of the seven Engineering Analysis Workstations (EAWs) used for console operations in the COBE MOR. The Applications Processor (AP), which is the computer that processes COBE telemetry for display in the MOR, will provide the CLEAR system with COBE telemetry, TDRS performance parameters and network information. Using this data, CLEAR will monitor the COBE-TDRS communications link in the search of problems and advise the analyst how to correct them.

CLEAR will be the first real-time diagnostic expert system utilized in a control center to support operations at NASA Goddard Space Flight Center. Currently, spacecraft communication links with the Tracking and Data Relay Satellite are used routinely and are planned to be utilized even more frequently and extensively by upcoming missions. This will provide a high degree of utility of the technology introduced by the CLEAR system.

System Requirements

The following are some of the functional and performance requirements specifications for CLEAR.

CLEAR is to have no effect upon the Application Processor (AP) and is to be transparent to other systems in the control center. The CLEAR system will be a strictly passive component of the system supporting COBE real-time operations. As an advisory, diagnostic expert system, CLEAR will monitor real-time data in an attempt to isolate problems, providing advice on how to correct them when they occur.

CLEAR is to be transportable within the COBE POCC. The system will run on any Engineering Analysis Workstation in

the COBE POCC without hardware modification and with the same operating system level software, e.g., communications package, graphics routines and device drivers, used by other application programs on the EAW. The workstations are AT-class personal computers running DOS and using non-standard graphics cards that support the Intelligent Systems Corporation (ISC) video format for compatibility with the POCC display systems.

CLEAR is to use the standard communication package developed for POCC workstation applications. The data furnished by the AP will be ordered ASCII text. The system will extract the TDRSS performance data, Operations Data Messages (ODM), and spacecraft status parameters from the communication buffer and convert them to the internal format required by the expert system.

CLEAR is to allow the operator to input COBE and TDRSS configuration parameter values for the subsequent TDRS support. CLEAR will utilize these parameters to identify misconfigurations of the communication systems in which case the system will notify the analyst of such discrepancies.

CLEAR is to be driven by ODM and status data sent by the AP. The expert system will monitor real-time Network Control Center (NCC) Operations Data Messages (ODMs) and TDRS and COBE performance parameters. When CLEAR isolates a problem, it will advise the operator how to correct it. The system will also monitor the input data frequency and will warn the analyst if data is not received within the expected interval (3 to 5 seconds).

CLEAR is to diagnose the faults identified during an event. The system will determine possible sources or causes of a fault, rank multiple possibilities in order of probability and present the results to the analyst. The system will also recommend the proper actions necessary to correct the problem. If requested, the system will explain why it believes that the fault exists.

CLEAR is to log all expert system activity for post event analysis. The system will time tag all identified faults and will record the inferences, the diagnoses, the recommendations offered to the operator. The system will provide non-realtime utilities to print a formatted copy of the log, to trace and analyze the activity of the expert system during the event and to extract statistics for evaluation of system performance.

CLEAR is to operate in real-time with a performance requirement derived from the expected 3 to 5 second communication buffer (input data) arrival frequency. The expert system will convert input data, check parameter values and perform inferences within this time interval. Event logging, operator dialog and explanations are not real-time events subject to the performance requirement.

Tool Selection

The real-time response required of the CLEAR system translated into a performance requirement for the expert system. The data driven and diagnostic nature of the expert system placed interface and inference logic requirements on the tool selected to build the application. Further selection criteria came from the hardware and software compatibility requirements.

A number of secondary (desirable rather than mandatory) requirements also used in the selection included cost, number of tool users, length of tool usage, stability of supplier, development environment and availability of source code. The secondary selection criteria were used to rank the expert system building tools that satisfy the mandatory requirements.

At the time of selection, several commercially available expert system building tools met the mandatory requirements based upon available information including independent benchmark tests, first-hand experience and product reviews. However, none was ranked higher than the 'C' Language Integrated Production System (CLIPS).

CLIPS, a tool for the development of expert systems, was created by the Artificial Intelligence Section of the Mission Planning and Analysis Division at NASA/Johnson Space Center. CLIPS is an inference engine and language syntax which provide the framework for the construction of rule-based production systems.

CLIPS was entirely developed in the programming language 'C' for performance and portability. The key features that attracted us to CLIPS were:

- Forward Chaining Rules
- Portability
- Satisfactory Performance
- Provision of Source Code
- Completely Integrated With 'C'
- Extensibility
- Fully Documented

The CLEAR development team is quite pleased with CLIPS. It suits the requirements of the system nicely, has been easy to integrate with the data and user interface subsystems, and has demonstrated respectable performance. The provision of the source code with the tool turned out to be quite useful. It provided the capability to modify the tool to accommodate unanticipated needs of the system thus avoiding the wasted time and effort required to switch to another expert system shell.

System Operation

To facilitate ease of use and to reduce demands on the analyst, user input is minimal. The only user input required is the pre-event initialization of parameters in the Configuration Table display (provided by the CLEAR system). Default settings are provided for each parameter and, to further expedite the process, a menu of common event configurations, called "prototype-event" codes, are provided in a menu-type format. Hence, the user only needs to select a prototype-event code and enter the event start-time in order to configure the expert system for operations.

The CLEAR user interface (figure 1) utilizes textual and graphical output in a windowed format to provide the analyst with information about the status of the communications links. The graphics window in the top left of the screen displays the current status of the communications links and elements of the communications network between the COBE spacecraft and the POCC. If the parameters indicate that a link or processing system is degrading or down, the associated icon will turn to yellow or red, respectively, thus providing the analysts with a reference of the status of the communications event in a quick glance.

When the expert system isolates a problem, a description of the problem will be displayed in the "Problems" window with the associated advice displayed in the "Advice" window. If

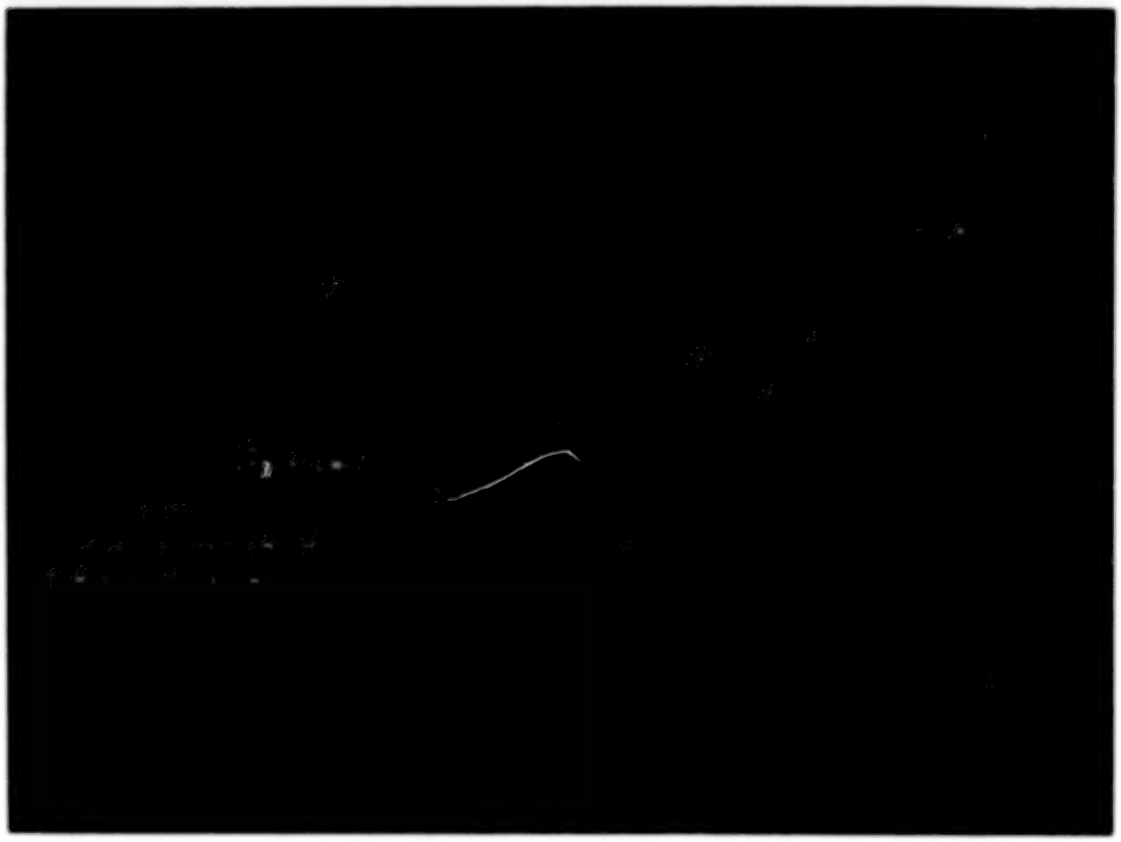


Figure 1. - Photograph of the CLEAR User-Interface

there are multiple problems, they are displayed from top to bottom in order of criticality. By default, the advice of the most critical problem is displayed in the Advice window; however, the system can display the advice to any problem that the analyst selects.

To assist the analyst and to promote system credibility, the CLEAR system provides an explanation facility. When commanded to do such, the expert system will replace the Advice window with a static explanation of why it believes that a specified problem exists. No backtracking or backward chaining is conducted since the expert system must continue to monitor the data in real-time to search for problems. Also, the analyst can view the display page of raw data from which CLEAR obtains its data. This can be used by the analyst to verify the expert system's results in the manner that he/she would have previously used.

Throughout the communications event, the CLEAR system logs data and problems isolated in an "event log". At the conclusion of the real-time support, the event reporting utility automatically converts the log into a more detailed and readable "event report". Also, a history file which contains a listing of the problems isolated in each event and a summary of the total number of occurrences of each problem for the mission to date is updated. The event report is useful for verification and debugging purposes and the history file provides information that allows the identification and analysis of problem trends which can be used to enhance both the knowledge base and operational support.

Development Approach

The CLEAR Expert System was developed in three phases: a rapid prototype phase, an operational prototype phase and an operational development phase. The rapid prototype phase was executed on a Lisp Machine using a high-powered, hybrid expert system development shell. In this phase, knowledge acquisition, requirements specification, and experimentation with knowledge representation schemes were begun. The system was demonstrated using simulated data provided by a data simulator software program that ran on a mainframe and was developed by the CLEAR team.

The operational prototype phase was conducted to further investigate the technical feasibility of the system within the functional requirements of the operational platform. The system was fully redesigned and developed using the programming language 'C' and CLIPS. The performance of the system in the run-time environment was investigated using data transmitted by the simulator at rates anticipated in operations.

The third and final phase involved the integration, testing and evaluation of the system in the COBE POCC. The system has a physical interface with the computer from which the expert system receives its data. CLEAR was tested using simulated data from the data simulator used to test all operational software developed for the POCC. Finally, the expert system was demonstrated and further tested using "live" data during spacecraft communication simulation events (used to prepare the Flight Operations Team in commanding and controlling the spacecraft). Full operational use will be realized after launch of the COBE spacecraft which, at the time of this writing, is scheduled for no earlier than mid-November, 1989.

Lessons Learned

A discussion of lessons learned, based on a retrospective analysis of the CLEAR development experience, is presented in this section. These points, and others, are more fully discussed in the paper "Integrating Expert Systems into an Operational Environment"2.

Involve the user. Involvement of the end-user in the evaluation process of computer systems is widely recognized as an important factor for attaining the functional goals of the system. This involvement seems to be more important for expert systems because, for an expert system to be successful, the users must have confidence in the problem solving capabilities of the system in order to fully accept it as a tool that will enhance their productivity, consistency, etc. This is highly unlikely if a system is brought in from "outside" and they are required to use it. When users are allowed to contribute in the development of the system, not only will the system more likely meet their needs, but they will also have greater faith in the system thus ensuring wider acceptance.

For many expert system development teams user involvement only consists of consulting users for feedback late in the development process. However, if users are included throughout the development process, many benefits will result. For instance, early involvement will emphasize their importance in the success of the project and, hence, they will actually feel like integral members of the development team. And, as integral members of the development team, not only will they strive harder for the success of the system, they will also become system advocates who will promote the system to others. This is valuable in situations where the user group is widely distributed or so large that it is unreasonable to involve all of them in the development process. Perhaps more importantly, an involved user will become a credible advocate when selling the project to management.

Finally, involvement of the user will also help control the cost of development since the user will be able to identify undesirable aspects of the system early in the development phase. As with any software project, the sooner problems are recognized and corrected, the less impact they will have on the project schedule and the less costly it will be to correct them.

Provide a robust software development environment. All too often the prototyping phase is conducted with a rich development environment while the development environment of the operational system is not nearly as supportive. The flexible development environment of the prototype is chosen to facilitate rapid coding and easy experimentation thus allowing speedy development of a system prototype. However, upon completion of the prototype, the development team mistakenly may assume that the problem domain is fully understood and the system architecture is completed. This is seldom the case. Changes to the knowledge base and system design will be necessary throughout the development of the operational phase, even after delivery. Many changes will result as the knowledge base and the user's needs evolve. Some people will only give the system full consideration during the development of the operational version when they realize that the system really is going to be used in operations.

Not only will the choice of a fully supportive software development environment enhance the development process, it will also greatly assist in the maintenance phase. Software maintenance, now considered an integral element of system development by software engineers, is often over-looked by expert system development teams. It is not uncommon for the maintenance of conventional software systems to cost up to 50% of the original development effort4. This figure is greatly increased by a lack of an accurate design specification and robust development environment (that supports easy debugging). Failure to provide a flexible software development environment for the development and maintenance of the operational version of an expert system is extremely short-sighted and often crippling to the success of the project.

Conduct a thorough prototyping phase. Success of an operational expert system is contingent upon a thorough prototyping effort. The purposes of the prototyping phase should be to:

- understand the problem domain,
- acquire the knowledge,
- identify appropriate knowledge representation schemes,
- establish rule precedence and exception handling,
- devise a suitable system architecture,
- define the requirements specification,
- analyze the business justification of the project,
- sell the project to management and the users.

Typically throughout the iterative process of prototyping, many of these critical purposes are not addressed thus severely jeopardizing the success of the project. Although this phase is often viewed as an experimental process, it should be conducted as an investigatory/ preparatory phase necessary for the proper development of an expert system intended to support operations.

The prototype must be demonstrated to management for approval and to the users for acceptance and feedback. Problems can result. It is a mistake to demonstrate a prototype with an underdeveloped user-interface (maybe consisting of textual output, and windows for the editor, debugger); it is equally wrong to demo a system that attempts to incorporate any and every function related to the problem solving process. An underdeveloped user-interface will likely appear complex or unfriendly, thus failing to secure approval for continued development. On the other hand, an overdeveloped user-interface can cause two problems. First, during development, the team will find the actual implementation of such an over zealous prototype to be more difficult than anticipated, delaying the project and wasting funds. Second, if the functionality of the prototype is cut back during the development phase, the expectations of the users will not be met thus reducing their enthusiasm and support for the system.

A problem resulting from demonstrating a handsome, well-behaved prototype is that some managers may want to deliver it "as is". Although the system seems to be complete from the outside, the developers realize that in the rush to produce a working prototype, overall software quality and long-term maintainability issues were not considered. Delivering a system in this state would not only be a software maintenance engineer's nightmare, but the resulting lack of robustness would rightfully scare away users and, even worse, convince operations managers that expert systems are too unreliable to support anything outside of the research labs.

A minor problem while prototyping is the failure to record all problems encountered, solutions used and explanations why chosen. These records help tremendously during the design of the operational system and maintenance. More importantly, these records will help ensure a smooth development of the operational system by avoiding the duplication of costly errors.

Thoroughly assess the effort required to interface the expert system to the user and the data source. A major issue usually focused upon in the prototyping phase is the acquisition, structure, and representation of the domain knowledge. Although this issue is important, it should not overshadow the necessity of conducting a complete system analysis to determine the amount of effort required to interface the expert system to the user and the data source. As with other software development projects, this figure is often grossly underestimated.

An accurate estimation of the effort required to develop the interfacing subsystems is best accomplished by a system analyst who has an understanding of the expert system, the data source, and the dynamic nature of an expert system development project. The knowledge engineer usually assumes this role; however, if he has limited experience in system analysis, the project's success can be endangered.

Test with "real" data as early as possible. Many groups develop an expert system that must interface with other systems for data without testing with "real" data until the final phases of development hence relying on the data to perfectly match its specifications as documented. This is a mistake because real world data is often incomplete and inaccurate.5 Slight variations from the specs can create potentially serious problems during integration and testing. Again, the earlier problems are found, the better off the project.

If access to real data is not feasible, develop a data simulator to facilitate testing. Although this sounds like a waste of time and money, the effort expended to develop one will not only be recouped through development, but it will also increase the quality of the system. Testing can be further augmented by the development of test suites with drivers to administer the tests automatically, logging the results into a file for debugging, maintenance, and record keeping. For large systems involving many programmers, automated test drivers will ease the laborious testing procedure that must be conducted after each modification.

Allow an independent group to test the system. If resources allow, arrange for an independent group to test the system. They will usually uncover a high proportion of the bugs due to their unbiased position and the fact that they have been challenged to discover as many bugs and weaknesses as possible.

Software testing is a critical element of quality assurance that is often executed in an unorganized and haphazard manner. The earlier bugs are isolated, the easier and less costly the removal process will be. Bugs found late in development may force major design changes and hence, increase development costs and delay system delivery. Even worse, if a user isolates a bug, his faith in the system will be diminished thus jeopardizing acceptance of the system.

Conclusion

Over the past few years, expert systems have emerged from the research labs to enhance a diverse array of operations. One of the prime areas of application of this maturing technology is in the area of real-time fault-isolation and diagnosis where the synergistic combination of the speed and tireless attention of the computer and the problem-solving knowledge of an expert (embedded in an expert system) have created powerful systems.

The CLEAR system readily demonstrates how expert systems can relieve the analysts of the tedious and demanding chore of monitoring screenfuls of data so that their time can be dedicated to solving higher level problems. While CLEAR is confined to fault isolation within the COBE-TDRS communications link, the techniques used by this system can be applied to other functions within the POCC. However, the ultimate goal would be to develop a larger expert system or network of cooperating expert systems that could perform all of the functions in the control center.

Acknowledgements

I would like to thank Walt Truszkowski, Dolly Perkins and Bob Dutilly for their assistance in preparing this paper.

References

1. Cholawsky, E.M. "Beating the Prototype Blues." in AI EXPERT San Francisco, Vol. 3, No. 12, December, 1988, pp. 42-49.

2. Hughes, P.M., "Integrating Expert Systems into an Operational Environment." in Proc. for Computers in Aerospace VII Conference, Monterey, CA. 1989.

3. Hughes, P.M., and Hull, L.G. "CLEAR: Communications Link Expert Assistance Resource" in Proc. 1987 Goddard Conference on Space Applications of Artificial Intelligence and Robotics, Greenbelt, MD: 1987.

4. Pressman, R.S. Software Engineering- A Practitioner's Approach. McGraw-Hill, Inc., New York, NY: 1987.

5. Smith, D.L. "Implementing Real World Expert Systems." in AI EXPERT San Francisco, Vol. 3, No. 12, December, 1988, pp. 36-41.

Process and Information Integration via Hypermedia

David G. Hammen
Daniel L. LaBasse
Robert M. Myers

The MITRE Corporation, Houston, Texas

Abstract

Success stories for advanced automation prototypes abound in the literature but the deployments of practical large systems are few in number. There are several factors that militate against the maturation of such prototypes into products. This paper addresses one issue, namely, the integration of advanced automation software into large systems.

Advanced automation systems tend to be specific applications that need to be integrated and aggregated into larger systems. Systems integration can be achieved by providing expert "user-developers" with verified tools to efficiently create small systems that interface to large systems through standard interfaces.

This paper explores the use of hypermedia as such a tool in the context of the ground control centers that support Shuttle and space station operations. Hypermedia can be an integrating platform for data, conventional software, and advanced automation software in hypermedia enables data integration through the display of diverse types of information and through the creation of associative links between chunks of information. Further, hypermedia enables process integration through graphical invoking of system functions. Through analysis and examples we are able to illustrate how diverse information and processing paradigms can be integrated into a single software platform.

Integrating Small and Large Systems

Small Systems Integration

Success stories for advanced automation prototypes abound in the literature but the deployments of practical large systems are few in number. There are several factors that militate against the maturation of such prototypes into products. One factor is that advanced automation systems tend to be specific, relatively small, applications that need to be integrated and aggregated into larger systems.

Application software that supports small groups of users or narrow disciplines is sometimes viewed as a threat to large systems. That is, severe controls imposed by system developers and maintainers of large systems (of necessity to ensure system integrity) are impediments to the development of small systems. By providing the user with verified tools and standard interfaces, systems integrators can enable the development of small systems that integrate well into large systems.

We perceive a need for a defined transition point between the systems integrators and the users that allows the users to automate their operations themselves. By providing a verified toolkit to expert users ("user-developers"), we can enable the user to streamline local operations without generating new requirements or change requests against the large system. That is, authority and responsibility for localized changes which affect small groups of users can be accomplished by small groups within the constraints of the toolkit and their management. This does not imply that configuration management (CM) should be abandoned for small systems. On the contrary, CM is a shared responsibility of the system developers and the users.

Systems Integration in Large Systems

A large group of people is needed to perform the systems integration task for a system as complex as a spacecraft ground control center. The users of small systems are usually not interested in, or qualified for, this large systems integration job. The systems integrators focus on widely shared system functions such as operating systems, data distribution, user

interface management, and configuration management. As a widely shared resource, an automation tool would be procured, verified, and tailored for control center use by the systems integrators. In addition to these verified automation tools, the user-developers must be provided with standard interfaces to the large system.

The large system into which advanced automation techniques will be incorporated for the Mission Control Center Upgrade (MCCU) can be best described as a "hardware independent software environment" or HISE. This is a run-time environment characterized and defined by industry standards. The standards that compose the HISE are the UNIX System V Interface Definition (SVID) for operating system services, X-Windows for the user interface, the Graphics Kernel System (GKS) for graphics, the Open Systems Interconnection (OSI) protocols for networking, and ANSI 'C' as the high level language. HISE manifests itself to the developer as libraries of function calls which provide services in an agreed-upon interface. The system is hardware independent in that the system need only provide the interfaces and implement the services as agreed upon in the standards. The HISE is a Level C requirement of the MCCU 2.5 delivery [JSC 1988].

The next section explores the use of hypermedia as one such automation tool that can be provided to user-developers.

Information Presentation via Hypermedia

Introduction

The term "Hypertext" describes a unified information system for non-sequentially accessing text and/or graphics. "Hypermedia" expands these concepts to include other media such as sound, photographs, and video. Hypermedia can be further extended. This paper investigates the possibility of using hypermedia as an integrating platform for executable processes as well as for information.

The use of hypermedia is well suited to needs of spacecraft ground control center where non-sequential access to large amounts of diverse information is the norm. In addition to their role as a sophisticated user community, flight controllers are also a sophisticated development community. Hypermedia meets well the development needs of a user community that wants to automate their operations themselves.

Visionary views of hypermedia typically involve far-reaching, ad-hoc searches of large collections of data by scholarly researchers [Bush, 1967; Nelson, 1987]. These visionary views are not particularly applicable to the more controlled environment of a manned space flight control center; they may even be seen as exemplifying an unacceptable technology.

Flight controllers use large amounts of diverse but linked information, but this use is more regimented than the scholarly access of data that is exemplified in these visionary views [Johns, 1987]. Hypermedia is as applicable to this more regimented use of data as it is to scholarly research. Any electronic form of flight documentation must display diverse kinds of information such as text, graphs, and schematics; many hypermedia systems excel at this. Each piece of the flight documentation is linked to other pieces (such as rules and rationales) in a variety of ways, and the handling of these linkages is the premier characteristic of all hypermedia systems.

An Example

Flight controllers direct crew operations according to predetermined plans. The crew procedures that compose the plan can be displayed electronically in various forms, for example as a timeline or as a time-tagged list. A hypermedia system could allow a controller to select an upcoming procedure directly from the plan and display the steps that need to be followed to carry out the procedure. The controller could also follow hypermedia links to examine associated procedure rationale, constraints, and supporting material such as layouts of onboard switch panels, diagrams depicting the Shuttle's attitude, or schematics of onboard systems.

Paper vs. Screen

Flight controllers currently use paper documents in Shuttle Mission Control Center (MCC). These documents are now being produced electronically, yet the final product of this process is still paper documents. Although the electronic medium is preferable for the production of the product it apparently is not for the actual use of the data. While tradition may be partially responsible for the preference for paper, another reason for this choice is that paper is often the better medium [Yankelovich, 1985; Johns, 1989].

Books inherently provide contextual information that is not conveyed well by most electronic information systems. For example, placemarkers may be easily attached to paper pages; notes may be easily written on paper pages [Yankelovich, 1985].

Hypermedia implementations of information systems can provide these advantages of the printed media while accentuating the advantages of the electronic media. Contextual information can be

provided with graphical cues. Annotations and markers can be added to a hypermedia display. [Bernstein, 1988]

Potential Problems

There are some problems that must be solved before hypermedia can be accepted for use in the control center. Controllers often do not have the luxury of making the wide-ranging searches that are proposed by hypermedia visionaries. The links must be carefully constructed to aid the controllers in performing their jobs [Johns, 1987]. This places an extra burden on the organizers of the documentation (beyond the primary task of collecting and designing the information). The data for existing paper products typically is not chunked to suit the needs of a hypermedia implementation nor are the links well enough defined, further extending the burden placed on the organizers of the documentation [Raymond, 1987].

Finally, a hypermedia implementation in an operational flight control environment must overcome another advantage of paper not mentioned above. Paper is static and easily controlled [Yankelovich, 1985]. A hypermedia system for the control center must have configuration control capabilities.

Process Integration via Hypermedia

We propose that the hypermedia also be used to integrate executable processes with information. There are many ways to encapsulate the lower level invocations of operating systems functions within a higher level construct. Hypermedia systems not only can accomplish this encapsulation but can also link these encapsulations with related documentation.

Reducing Information Overload

Hypermedia can provide a graphical front end to system services [Conklin, 1987]. This could reduce manual activities (such as typing the name of a program and associated data files) and thereby help to keep the user's mind on the task being performed. As with the information integration system, the process integration system should be built by the users; only they know the processes to be performed and the links between them and the supporting documentation.

An Example

A controller may need to execute a specialized program to analyze suspect components in response to sensed anomalies. One way to do this is to tell the operating system to execute the program by typing in the name of the program and associated data files. Another way is to select a graphical icon for the program. This latter technique helps keep the user focused on the analysis. A hypermedia approach extends this technique by associating the icon with other objects that are related to the suspect components. The controller could transfer from the static hypertext documentation to the active analysis program by selecting the appropriate item from the hypermedia display that integrates the documentation with the process icon.

How to Proceed

The controllers' flight procedures and related material are excellent candidates for a hypermedia information access system. It is important to get the users involved in development and evaluation of such a system. Only the users know what the data is, how to "chunk" the data into meaningful information, and what links to establish between the chunks of information. The users of this system could also be the small systems builders, user-developers, performing the systems integration task using hypermedia as a systems integration toolkit. The systems integrators of the large systems need to support this activity by acquiring the hypermedia tool, verifying it for space-rated use in a control center, and providing it to the user-developers.

Summary

We perceive a need for a defined hand-off point between the systems integrators and the users that allows the users to automate their operations themselves. This can be achieved by providing user-developers with verified tools, such as hypermedia, by which they can efficiently create small systems that interface to large systems through standard interfaces.

Acknowledgements

We thank Ken Jenks and Mark Dean of Rockwell Shuttle Operations Co. for their suggestions on the applicability of hypertext to the flight data file. We thank Dona M. Erb of the MITRE Corporation who provided background information on hypermedia and critiqued this paper.

This project is a part of NASA Research, Technology, Objectives and Plans (RTOP) task 488-50 and is administered by the Systems Development Division at the NASA Johnson Space Center.

References

Bernstein, M. (1988), *The Bookmark and the Compass: Orientation Tools for Hypertext Users*, ACM SIGOIS Bulletin, Vol. 9, No. 4, October 1988, pp. 34-45

Bush, V. (1967), *Memex Revisited*, Science Is Not Enough, V. Bush, ed., New York, NY: William Morrow & Co., Inc., Apollo Edition, 1969

Conklin, J. (1987), *A Survey of Hypertext*, MCC Technical Report STP-356-86, Rev. 2, Austin TX: Microelectronics and Computer Technology Corporation, December 1987

Johns, G. (1987), *Flight Data File for the Space Station, Volume I: Baseline Definition*, MTR-10019, Vol 1, Houston TX: The MITRE Corporation, February 1987

Johns, G. (1989), *Dynamic Display of Electronic Crew Procedures for Space Station*, Journal of Spacecraft and Rockets, Vol. 26, No. 1, January-February 1989, pp. 39-46

JSC (1988). *Workstation Executive (WEX) Requirements for Delivery 2.5*. JSC-22363. October, 1988

Nelson, T. (1987), *Literary Machines*, Abridged Electronic Edition 87.1, Macintosh hypertext edition in GUIDE, Bellevue WA: OWL, International, Inc., 1987

Raymond, D.; Tompa, F. (1987), *Hypertext and the New Oxford English Dictionary*, HyperTEXT'87 Papers, Chapel Hill, NC: The University of North Carolina, 1987, pp. 143-153

Yankelovich, N.; Meyrowitz, N.; van Dam, A. (1985), *Reading and Writing the Electronic Book*, Computer, Vol. 18, No. 10, October 1985, pp. 15-30

AN APPROACH TO A REAL-TIME DISTRIBUTION SYSTEM

Frank P. Kittle, Jr.
McDonnell Douglas
16055 Space Center Blvd.
Houston, TX 77062

Eddie J. Paddock
McDonnell Douglas
16055 Space Center Blvd.
Houston, TX 77062

Tony Pocklington
McDonnell Douglas
16055 Space Center Blvd.
Houston, TX 77062

Lui Wang
NASA - JSC
FM72
Houston, TX 77058

ABSTRACT

The requirements of a real-time data distribution system are to provide fast, reliable delivery of data from source to destination with little or no impact to the data source. In this particular case, the data sources are inside an operational environment, the Mission Control Center (MCC), and any workstation receiving data directly from the operational computer must conform to the software standards of the MCC. In order to supply data to development workstations outside of the MCC, it is necessary to use gateway computers that prevent unauthorized data transfer back to the operational computers.

Many software programs produced on the development workstations are targeted for real-time operation. Therefore, these programs must migrate from the development workstation to the operational workstation. It is yet another requirement for the Data Distribution System to ensure smooth transition of the data interfaces for the application developers. A standard data interface model has already been set up for the operational environment, so the interface between the distribution system and the application software was developed to match that model as closely as possible. The system as a whole therefore allows the rapid development of real-time applications without impacting the data sources.

INTRODUCTION

In this real-time distribution system, the recipients are development programs supported by the Mission Planning and Analysis Division (MPAD). The data sources are controlled in an operational environment which is located in the MCC. The programs under development are real-time applications such as artificial intelligence expert systems. These applications require access to real-time data streams for development and testing phases. They are also targeted for real-time operational workstations inside of the MCC, so all data interfaces to the distribution system must provide for a smooth transition between these two environments.

Data is transferred to operational workstations throughout the MCC by means of Local Area Networks (LANs). Any workstation receiving data through these LANs must conform to the software standards and procedures of the MCC. It is therefore necessary to use Gateways as a common medium for supplying data to the development workstations outside of the MCC. These Gateway workstations adhere to the MCC standards and procedures and most importantly prevent unauthorized data transfers back to internal MCC systems, such as the Mission Operations Computer (MOC).

GATEWAYS

In the MPAD data distribution system, there are two operational Gateway machines receiving data from two separate LANs in the MCC (see Figure 1). The MITS (MOD-IPS-TACAN System) Gateway provides for trajectory data, and the CAS (Calibrated Ancillary System) Gateway provides for telemetry data. These Gateways transfer data through a common LAN to designated development workstations. This through-put connectivity is handled with User Datagram Protocol (UDP) sockets.

The UDP protocol provides a procedure for application programs to send data packets to other programs with a minimum of protocol mechanism. The protocol is transaction oriented, and delivery and duplicate protection are not guaranteed. This means that bad or missing data packets are not re-transmitted as with the Transmission Control Protocol (TCP). In the Gateway scenario, this re-transmission is not desirable due to the high data flow requirements imposed on the Gateway machines. The UDP socket connection also allows the flexibility of a multi-socket to single-socket connection which is used in the Gateway (MITS and CAS) to development workstation network interface. In this interface, each Gateway has a

separate "sending" socket that transfers data to a single "receiving" socket at the development workstation. Hence, the UDP protocol is ideally suited for the MPAD data distribution system.

Using Gateways provides the development users with real-time operational data streams while isolating their applications from the strict operational procedures enforced by the MCC. All applications running on an MCC operational workstation must be "flight certified" for each shuttle flight. Flight certification includes rigid configuration management and extensive integration testing for each software release, with all certified software being frozen 45 days prior to the launch date. The Gateway software is a flight certified application running on an operational workstation, whose job is to transfer data to workstations outside the MCC while not allowing access to the MCC operational systems. Since the certified gateways do not allow access from the development workstation back to the MCC, they relieve the development user from the certification requirements while providing him with the benefit of real-time data streams.

DEVELOPMENT WORKSTATIONS

The MPAD development workstations are UNIX platforms that provide application programs with a software interface to the real-time operational data sources. This interface can be described by dividing it into two software components: the Network Interface Driver (NID) and the Application Interface Layer (AIL) (see Figure 2). The NID interprets UDP data packets from the LAN and then routes the packets through a shared memory buffer pool system to message queues which are designated for each data source driver in the AIL. The routing destination is defined by the Data Source Routing Table. This table is configurable by programs running in the AIL. This allows for dynamic start-up and shutdown of real-time data sources on the development workstation.

The message queues mentioned above allow for a quick passage of shared memory buffer pointers from the NID to the AIL. Each message queue has a corresponding data source driver. These drivers are responsible for translating data types that originate from their respective sources to a common Real-Time Interface (RTIF) format. After translation, the driver transfers the data to another shared memory area and releases the passed buffer. This shared memory area is surrounded by an RTIF layer which, through a set of function calls, allows applications to connect to real-time data sources. The RTIF is governed by an Interface Agreement (IA) document that was developed by a working group for the MCC environment. This IA provides for the software commonality that allows for a smooth transition when the application migrates from the development environment to operational workstations inside the MCC.

OPERATIONAL WORKSTATIONS

The MCC operational workstations are also UNIX platforms which have operational LAN connections. These workstations have strict configuration management and LAN access procedures which are controlled by a Workstation Executive (WEX) software package (see Figure 3). All operational applications run under WEX, and connect to real-time data sources through interfaces provided by the executive's operational software. These interfaces, such as its RTIF, are software libraries with the same syntax and functionality as those used on the development workstations. Since these RTIF software libraries adhere to the same IA, applications can be ported to the operational environment with little effort. Porting requires transferring the application source, re-compiling and then linking in the operational RTIF library. Therefore, by using the RTIF, user applications see no difference in the data interface between development and operational machines.

INTERFACE SOFTWARE

All applications and interface software described in this distribution system are written in the C programming language on UNIX operating systems. The application interface software uses multi-processing with a heavy emphasis on Inter-process Communication Software provided by UNIX. The workstations described in this configuration are UNIX platforms from a wide selection of vendors. Portability of applications between these workstations and generic interfaces that support all data sources were the key driving forces, and were considered in all software design phases.

CONCLUSION

In summary, this approach to a real-time data distribution system provides development users outside of the MCC with an interface to MCC real-time data sources. In addition, the data interface was developed with a flexible and portable software design. This design allows for the smooth transition of new real-time applications to the MCC operational environment.

MCC Working Environment Hardware Diagram

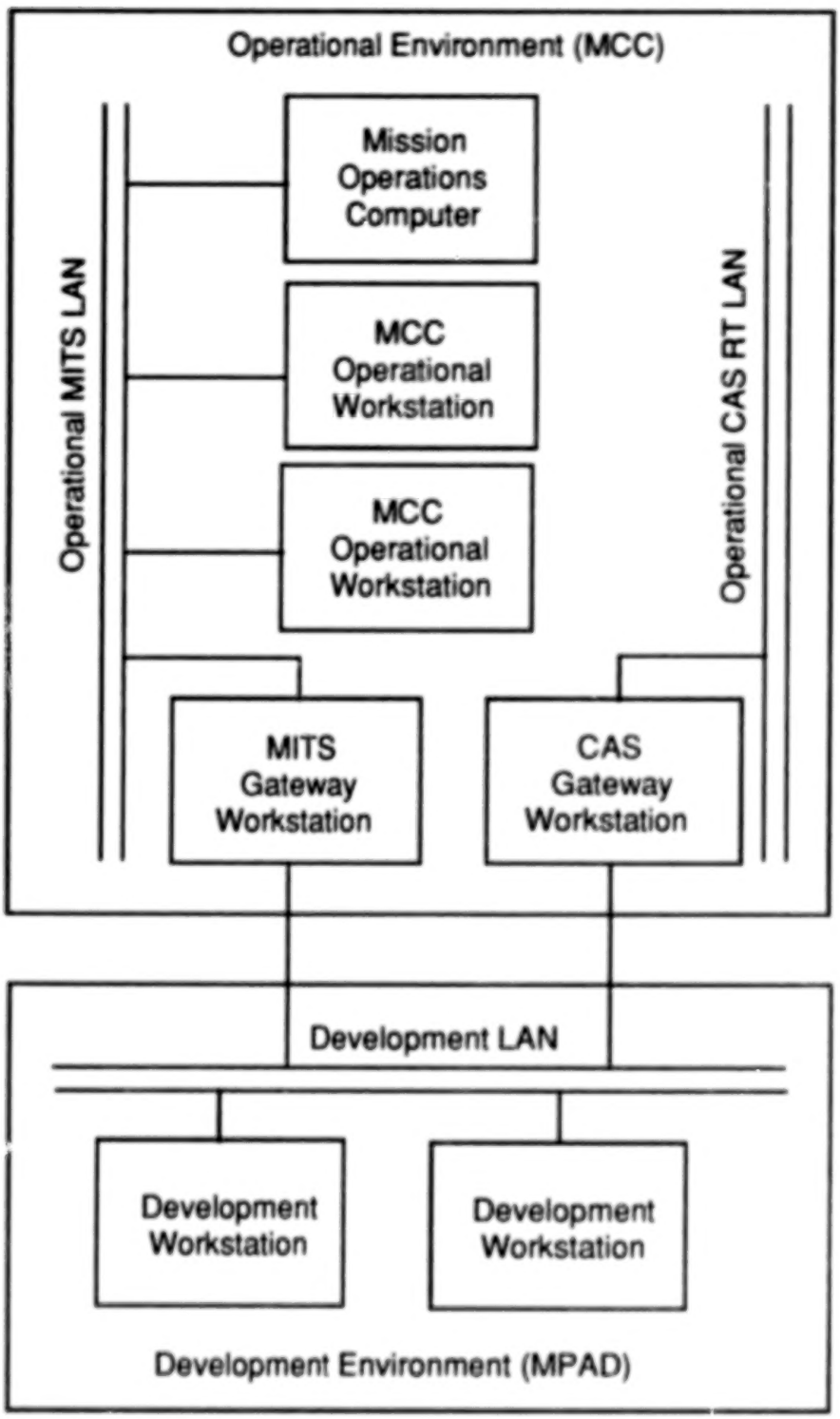


Figure 1.

Development Workstation Software Diagram

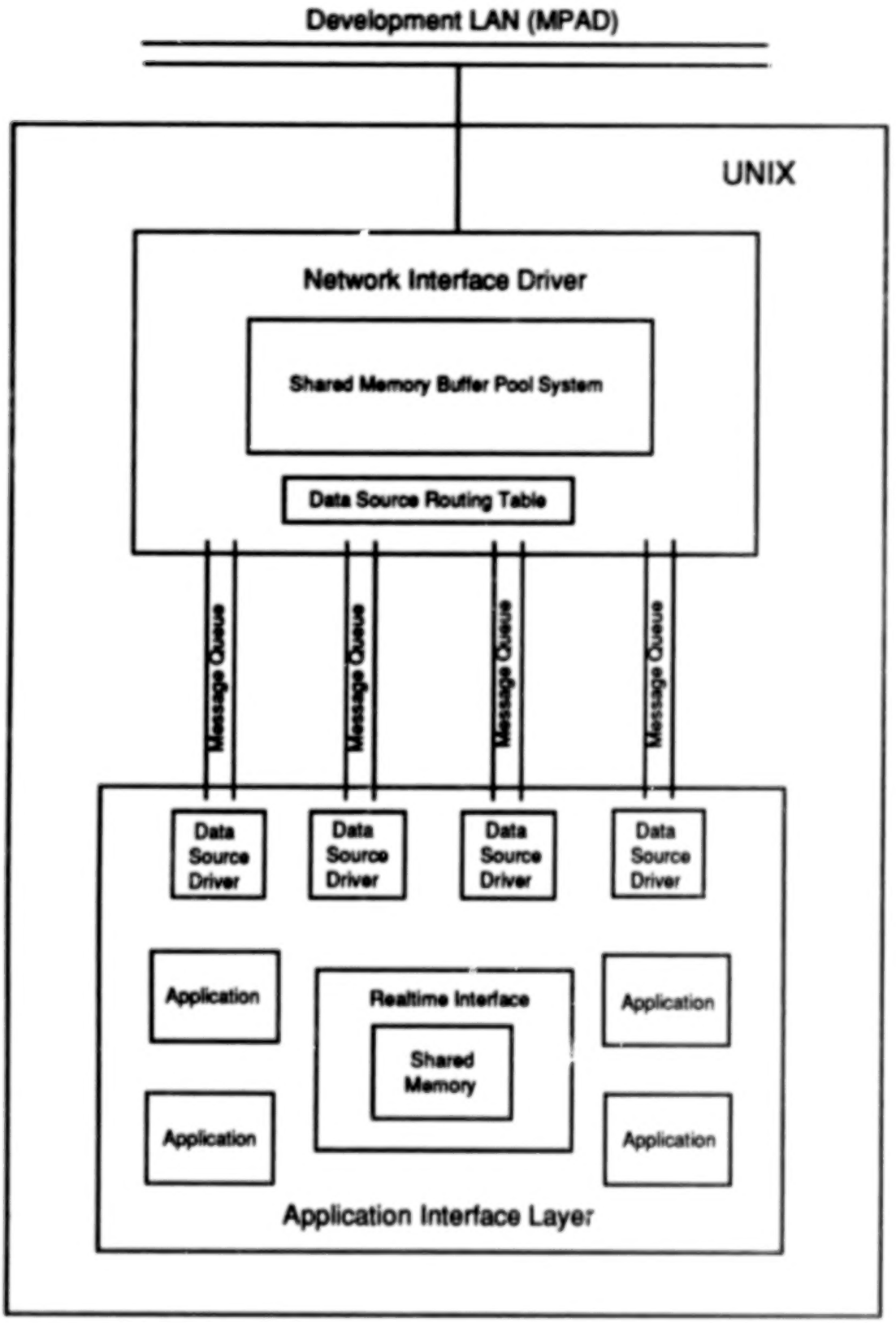


Figure 2.

MCC Operational Workstation Software Diagram

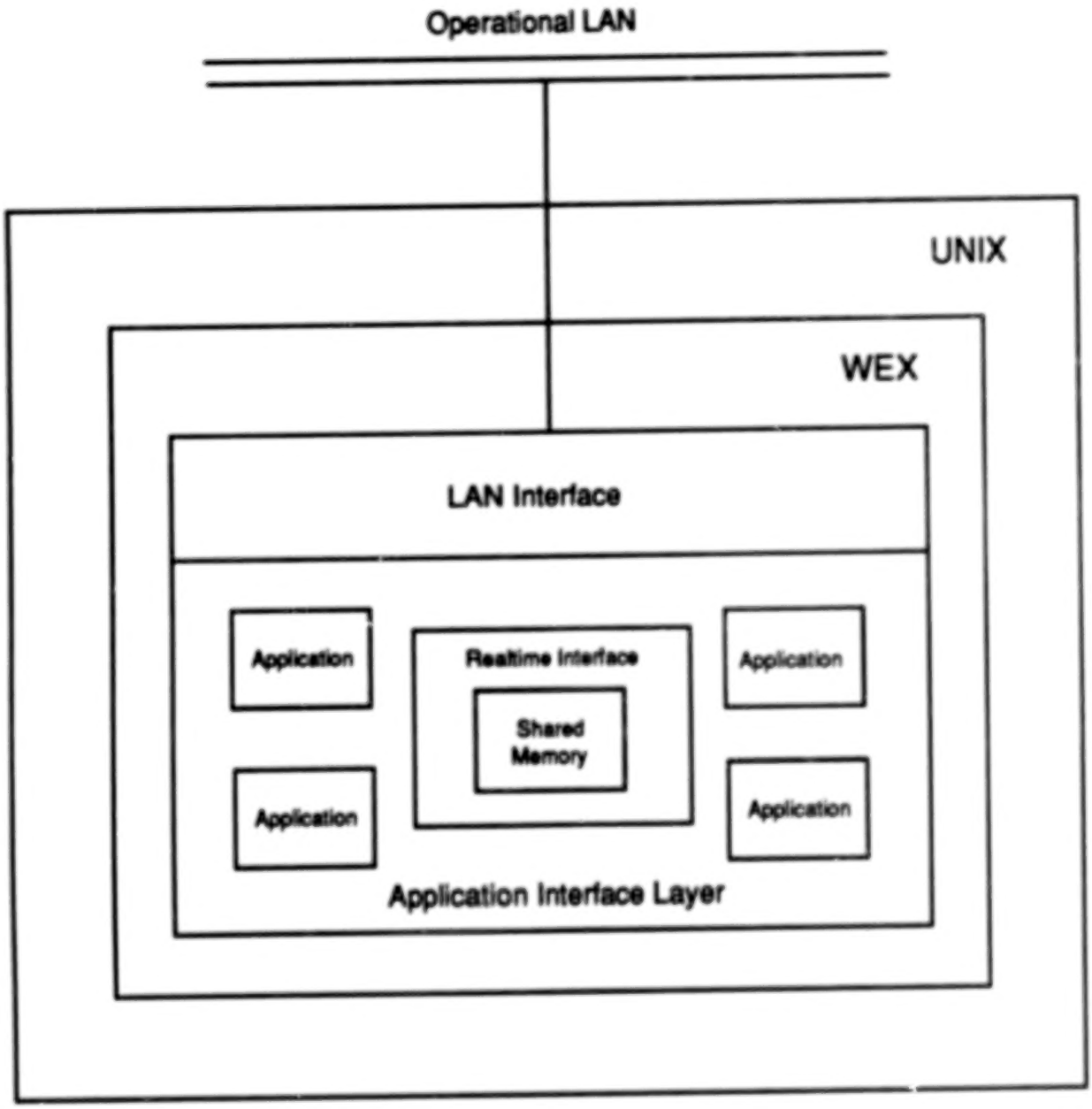


Figure 3.

A comparison of two neural network schemes for navigation.

Paul Munro
Department of Information Science
University of Pittsburgh
Pittsburgh PA 15260

munro@idis.lis.pittsburgh.edu

Abstract. Neural networks have been applied to tasks in several areas of artificial intelligence, including vision, speech, and language. Relatively little work has been done in the area of problem solving. Two approaches to path-finding are presented, both using neural network techniques. Both techniques require a training period. Training under the back propagation (BPL) method was accomplished by presenting representations of [current position, goal position] pairs as input and appropriate actions as output. The Hebbian/interactive activation (HIA) method uses the Hebbian rule to associate points that are nearby. A path to a goal is found by activating a representation of the *goal* in the network and processing until the current position is activated above some threshold level. BPL, using back-propagation learning, failed to learn, except in a very trivial fashion, that is equivalent to table lookup techniques. HIA, performed much better, and required storage of fewer weights. In drawing a comparison, it is important to note that back propagation techniques depend critically upon the forms of representation used, and can be sensitive to parameters in the simulations; hence the BPL technique may yet yield strong results.

Introduction

Description of the problem.

A *map* is given, which is a represention of landmarks and allowed paths and/or obstacles in the relevant region of space. Given an arbitrary pair of points, **I** (initial) and **G** (goal), the problem is to compute a sequence of actions which will bring the subject from **I** to **G**. Several approaches to this classic problem have been put forward (see for example, Brooks, 1983). These tend to rely upon explicit geometrical computations on polygonal representations of obstacles. In contrast, any geometrical considerations in the neural network approaches described below are *implicit*; that is, they are emergent artifacts of the learning processes. Two principles for processing and training of neural networks are briefly described in this section. More detailed treatments can be found in the references.

Back-propagation learning

Back propagation (Rumelhart, Hinton, & Williams, 1986) is a general algorithmic framework for training a feed-forward network of semi-linear units by randomly selecting pairs of input-output patterns from a training set and incrementally adjusting the network parameters, such that the network produces the appropriate output for a given input. The parameters of the network are usually, but not necessarily, restricted to the weights on the links (edges, in graph theoretic terminology) between the units (nodes). Initially, the parameters are set to random values. With each presentation of an input-output pair, the network produces a response to the input, which is compared to the desired output; the back-propagation learning (BPL) algorithm specifies a method for adjusting the network parameters, such that the discrepancy between the response and the desired output is reduced. The procedure is based on a gradient descent of the parmeter vector across an error measure. Like other gradient descent techniques, BPL is not guaranteed to find the global minimum; instead,

it often gets stuck in local minima, which may nevertheless result in acceptable performance by the network. As originally conceived, BPL was limited to static patterns; however there has been recent progress in processing time-varying inputs. (e.g. Jordan, 1987; Elman, 1988).

Hebbian/Interactive Activation

An interactive activation network (McClelland & Rumelhart, 1981) consists of a population of "neuron-like" elements, each representing an identifiable concept, in most implementations. The nodes are connected with positive weights, if their concepts are positively associated, and with negative weights if they are negatively associated. Normally, the weights are "hard-wired"; that is, the weights are preset and do not modify. However Hebb's postulate (1949) can be realized as a differential equation for learning in such networks, as has been done in other models, such as the Brain State in the Box (BSB) model of Anderson (1977).

A BPL approach to navigation: Method and results

The back propagation algorithm is typically applied to categorization problems, by learning an input-output mapping, where the inputs are exemplars and the outputs are categories. Jordan (1987) showed how a network could be trained to learn sequences, by partitioning the input into a represention labelling the sequence and a representation of one element of the sequence, and the output as a representation of the successor element in the sequence. Below, a similar scheme is applied to the navigation problem. The input is partitioned into a representation of the current state and a representation of the desired (goal) state. The output drives some sort of effector which changes the current state. The network architecture is shown in Figure 1. The current state and goal states are represented as patterns of activation across sets of input units. Each of the input units is connected to each of the "hidden" units (hidden, because they do not interact with the environment external to the network) in the next layer. Each of the hidden units is connected to each of the output units. The connection matrices are symbolized by the bold arrows in Figure 1.

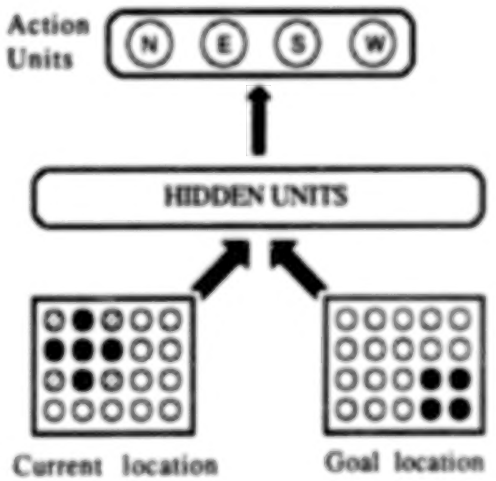


Figure 1. Basic route-finding network. Input units specify current and goal locations. The network generates an appropriate action as the output. The current location is updated according to the interaction of the action with the physics of the environment.

Attempts were made to train such a network on a very simple environment, consisting of a 5 by 5 grid of cells, each accessable by a single step from its 4 (N, E, S, and W) neighbors. With a small set of training data, the network was able to learn the steps in that set perfectly. However, if the set became too large, performance would

suffer. The performance of the network (with the addition of a second layer of hidden units) is shown in Figure 2. For each of the 25 possible goals, a five by five matrix of arrows depicts the motion taken by the network . A circle indicates where the position is identical to the goal; thus for each matrix the circle is the target. Note that while the trend is generally correct, the network makes errors that lead to dead ends (edges) or limitless oscillations (for example, when two arrows point toward each other).

```
O^<<^   <O<<^   <^O>^   >^>O^   >>>>O
>^<<<   >^<<<   >>^<<   >>>^^   >>>>^
^^<<<   ^^^<<   >^^^<   >>^^^   >>>^^
^^^<<   ^^^^<   ^^^^^   >^^^^   >>^^^
^^^^^   ^^^^^   ^^^^^   >^^^^   >>^^^

<<<<<   <<<<^   v^v<^   >>>>^   >>>>^
O<<<<   >O<<<   >>O<<   >>>O^   >>>>O
^<<<<   >^<<<   >>^<<   >>>^<   >>>>^
^^<<<   ^^^<<   >^^^<   >>^^^   >>>^^
<^^<<   ^^^^^   >^^^^   >>^^^   >>^^>

<<<<<   v<vv<   v>vv^   v>vv^   >>>>^
<<<<<   vv<<<   >>v<<   >>>v<   >>>>>
O<<<<   >O<<<   >>O<<   >>>O<   >>>>O
<<<<<   ^^<<<   >>^<<   >>>^^   >>>>^
<<<<<   v^^<^   v>^^^   >>^^^   >>^^>

v<<v<   vvvv<   vvvv<   v>vvv   v>vv>
vv<<<   vvv<<   >vvv<   >>vvv   >>>vv
v<<<<   >v<<<   >>v<<   >>>v<   >>>>v
O<<<<   >O<<<   >>O<<   >>>O<   >>>>O
<<<<<   v<<<<   >>^^>   v>^v>   >>>v>

vvvv<   vvvv<   vvvvv   vvvvv   v>vvv
vvv<<   vvvv<   vvvvv   >vvvv   >>vvv
vv<<<   vvv<<   >vvv<   >>vvv   >>>vv
<<<<<   vv<<<   >>v<<   >>>v<   >>>>v
O<<<<   v0<<v   v>O>v   >>>O>   v>>v0
```

Figure 2. The results from a fully connected BPL network. See text for description.

It was generally found that learning was much better when the patterns were presented to the network independently of the previous pattern. In the initial investigations, a particular goal was held constant while the network was trained on a sequence of steps leading to the goal, after which the goal was shifted to a random location. However, under such a training schedule, the goal position can remain constant too long, such that the weights from the current position representation "forget" what they learned with respect to other goals.

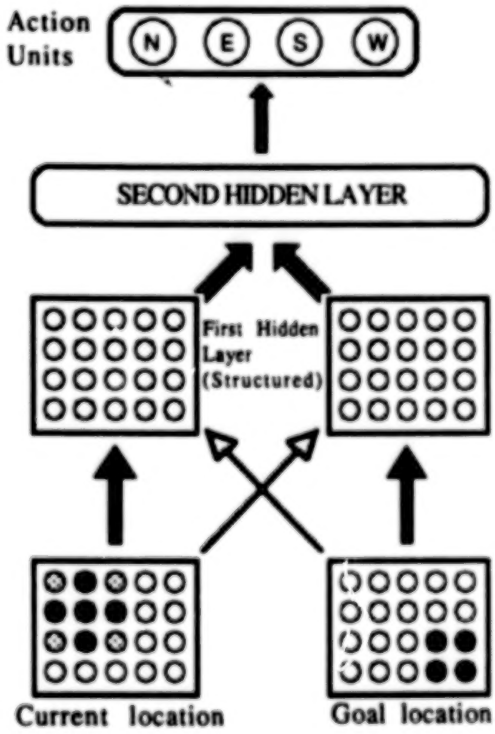


Figure 3. In this network, two sets of units have been inserted between the input layer and the hidden unit layer. One has a set of one-to-one connections with the GOAL input units and is fully connected to the CURRENT input units, and the other is connected in a complementary fashion. This "structured" hidden layer facilitates learning, but leads to poor generalization.

To remedy this, a more complex architecture was introduced (see Figure 3), by inserting another layer of hidden units into the previous structure between the input layer and the hidden layer. This new hidden layer consists of two sets of units. One set has one unit corresponding to each unit in the goal location input layer and receives input from that unit alone among the goal location input units; all of the units in that set receive input from *all* units in the

current location input set. That is, that set in the first hidden layer receives one-to-one connections (thin arrow) from the goal location input set and is fully connected to the current location input set (thick arrow). The other hidden set has a complementary set of connections. The one-to-one connections were *gates*, or *multiplicative connections*; that is, unless input was received from one of these connections the hidden unit did not respond.

With this architecture, the network was able to learn the 5 by 5 environment perfectly, as shown in Figure 4. However, in this case learning is quite brittle. The network is now nothing more than a lookup table, since it has specific weights corresponding to every input combination. Thus, there is no generalization of information from one learning trial to any other situation.

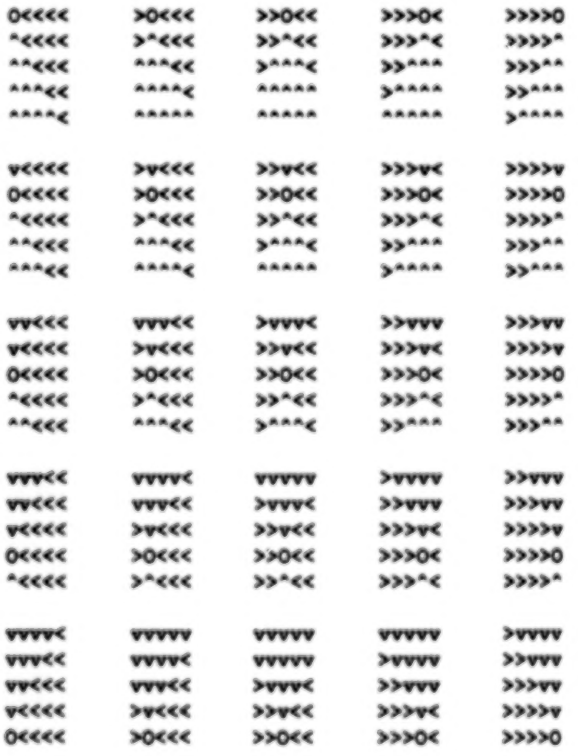


Figure 4. The results from a partially connected BPL network. See text for description.

A Hebbian/Interactive activation approach to navigation: Method and results

In this network, the environment was represented similarly to the above input representation, in that there is a unit for every landmark in the environment. Again, in consideration of designing the computer simulation, a rectangular grid was used. However, the architecture was quite different. In this network, all units were connected (initially) to all other units. Simulations using this model were performed in more complex environments; here, not every grid element was connected to its four neighbors. Instead environments, such as that shown in Figure 5, were used.



Figure 5. An example of a maze environment, such as was used in the HIA simulations.

Training was accomplished by an exploratory "wandering" process through the maze. Each cycle of the simulation began with taking a random step from the current position to a neighbor (neighbors in this case were defined by the links in the maze), with no backtracking unless necessary. Upon arrival at a node, the corresponding unit in the activation network was activated. Activation in each unit would decay by a fraction α, with each cycle of the

simulation. The weights bewteen units would increase in proportion to the product of the activity in the two units, and decay by another factor β. Appropriate choice of α and β led to a situation in which the weights between adjacent units were much stronger than the weights bewteen units two or more steps removed. Hence, these were all set to zero, and the neural net became isomorphic to the maze. This network was used to compute paths between arbitrary points in the maze by the following three stage procedure:

[1] The unit, *g*, corresponding to the goal is stimulated continuously at a high level, K, and activation spreads through the network via repeated iteration of matrix multiplication and simultaneous exponential decay, until the unit, *c*, corresponding to the current position is activated to a criterion level θ:

$$A_j(t) = \sum_k W_{jk} A_k(t) - \eta A_j(t) \quad \text{for } j \neq g$$

$$A_g(t) = K \text{ until } A_c > \theta$$

[2] The resulting pattern of activation **A** is then multiplied, element by element, by the pattern of the squares of the weights connecting *c* to the other units, $\mathbf{W_c}$.

[3] The current position is then updated, by moving to the unit with the greatest resulting product:

$$c(t+1) = \text{index } j \text{ that gives a maximum for } W^2_{c(t)j} A_j$$

Steps [2] and [3] are repeated until the goal is reached.

This method was found to work quite well over a set of different mazes, usually finding the shortest path. In cases where the shortest path was not found, the result was close to the optimum.

Discussion

While BPL was found to be inadequate for solving relatively simple problems, it should be recognized that it frequently requires considerable time and effort (and educated guesswork) to apply it successfully to a particular problem. The pattern representations must be carefully considered. Also, the network architecture and even such parameters as the learning rate and the momemtum (see Rumelhart, Hinton, and Williams, 1986 for a detailed description) can be critical in determining the success of a particular simulation. Thus, while the results reported above are discouraging, it is too soon to dismiss this approach.

The second technique, HIA, performed much better. The algorithm for finding a path is not especially novel; it is essentially equivalent to searching though a graph for the shortest path. The novelty is in the Hebbian modification technique used to construct the graph via temporally correlated activitations. This is somewhat sensitive to the parameters α and β. Further work is required for a general solution using this approach.

Other future plans include using more sophisticated representations for location, using multiple maps of the environment, such as maps for various types of transportation (e.g., walking vs. driving), or maps covering various scales (e.g., city maps vs. world maps). Recent work (Munro & Hirtle, 1989) has shown how the interactive activation model can account for a variety of documented psychological data, which indicates interactions between internal representations of different maps in free recall of geographical information. Conceivably, a hybrid technique, involving both the BPL and HIA methods will be used. Such a combination would probably use HIA for the high level planning and BPL to issue the action commands to the drive mechanism.

References

Anderson, J. A., Silverstein, J. W., Ritz, S. A., and Jones, R. S. (1977) Distinctive features, categorical perception, and probability learning: Some applications of a neural model. *Psychological Review*, 84:413-451.

Brooks, R. (1983) Solving the find-path problem by good representation of free space. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC 13: 190-197.

Elman, J. L. (1988) Finding structure in time. CRL TR 8801. Center for Research in Language. University of California at San Diego.

Hebb, D. O. (1949) *The Organization of Behavior*, New York: Wiley.

Jordan, M. I. (1987) Attractor dynamics and parallelism in a connectionist sequential machine. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, 531-546.

McClelland, J. L. and Rumelhart, D. E. (1981) An interactive activation model of context effects in letter perception: Part I. An account of basic findings. *Psychological Review*, 88:375-407.

Munro, P. and Hirtle, S. C. (1989) An interactive activation model for priming of geographical information. *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, [Accepted for presentation]

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986) Learning internal representations by error propagation. In: *Parallel distributed processing: Explorations in the microstructure of cognition*. D. E. Rumelhart and J. L. McClelland, eds. Cambridge, MA: MIT

THE FLIGHT TELEROBOTIC SERVICER (FTS)
NASA'S FIRST OPERATIONAL ROBOTIC SYSTEM

J. Andary, K. Halterman, D. Hewitt, P. Sabelhaus
Goddard Space Flight Center
Greenbelt, Maryland 20771

ABSTRACT

NASA has completed the preliminary definition phase of the Flight Telerobotic Servicer (FTS) and is now preparing to begin the detailed design and fabrication phase. The FTS will be designed and built by Martin Marietta Astronautics Group in Denver, CO, for the Goddard Space Flight Center, in support of the Space Station Freedom Program. The design concepts for the FTS are discussed, as well as operational scenarios for the assembly, maintenance, servicing and inspection tasks which are being considered for the FTS. The upcoming Development Test Flight (DTF-1) is the first of two shuttle test flights to test FTS operations in the environment of space and to demonstrate the FTS capabilities in performing tasks for Space Station Freedom. Operational planning for DTF-1 is discussed as well as development plans for the operational support of the FTS on the space station.

INTRODUCTION

Project Status

The Flight Telerobotic Servicer (FTS) project was formed in 1986 to develop a telerobotic device to perform assembly, maintenance, servicing and inspection tasks on the space station, the shuttle orbiter or the Orbital Maneuvering Vehicle (OMV). A phase A study was conducted in the fall of 1986 with agencywide participation [1]. The output of this feasibility study was a preliminary requirements document and a request for proposals (RFP) for competitive phase B studies to establish a preliminary design of the FTS. The phase B study contracts were awarded to Grumman Space Systems Division in Bethpage, NY, and Martin Marietta Astronautics Group in Denver, CO. These studies ran for 9 months and were concluded in September 1988.

An in-house phase B study was conducted by NASA in order to refine the requirements for the FTS and to begin working the interfaces with Space Station Freedom [2 and 3]. The in-house study was started several months earlier than the contractor studies, because NASA needed some early estimates of the FTS resources and operational capabilities to work interface issues with the Space Station Freedom Program, which had already completed its own phase B study and was preparing to start detailed design and implementation.

One of the outputs of the phase B studies was a new requirements document for the phase C/D. This became part of the phase C/D RFP which was released November 1 1988. Proposals were received on January 3, 1989, and Martin Marietta was selected for negotiations on April 20, 1989. This paper describes the proposed Martin Marietta design for the FTS, which is shown in Figure 1.

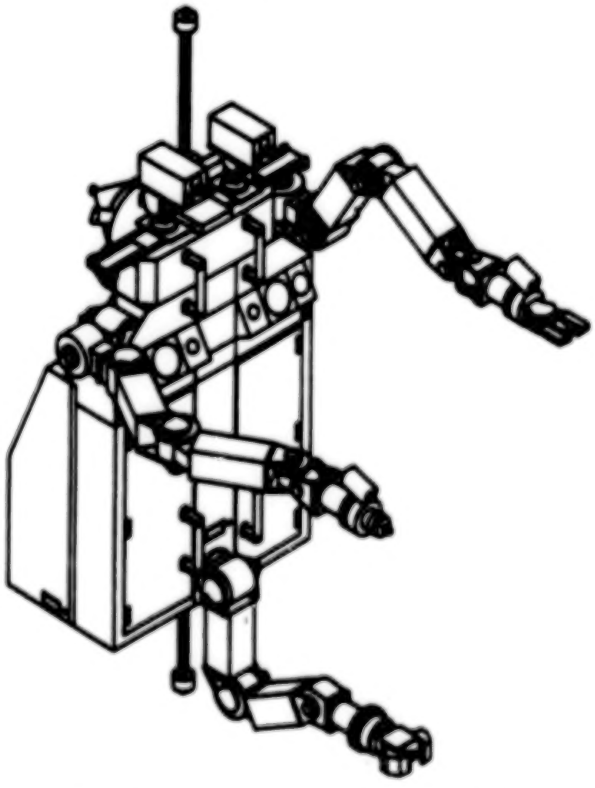


Figure 1. Flight Telerobotic Servicer (FTS)

Test Flights

Before describing the space station FTS (SSFTS), which will be launched on one of the early Freedom Station assembly flights, there are two early shuttle test flights which should be mentioned. They are the Development Test Flight (DTF-1), scheduled for launch in August, 1991, and the Demonstration Test Flight (DTF-2), scheduled for launch in 1993.

The DTF-1 will be flown as an attached payload to test the control and performance of the manipulators in zero gravity,

as well as the human-machine interface through the workstation. Test tasks will be performed, and data will be collected so that the performance on orbit can be compared tc the performance in the laboratory. The present concept of the configuration of the payload bay elements for DTF-1 are shown in Figure 2. The manipulators and the upper torso of the telerobot body are mounted to a Multipurpose Experiment Support Structure (MPESS) bridge. There will be one or more task boards attached to the MPESS. They contain the task elements which will be manipulated during the flight. An astronaut will teleoperate the DTF-1 from the workstation located on the aft flight deck of the orbiter with two minimaster hand controllers and television screens that display images from the four cameras that are located on the telerobot. There is also a fifth global-view camera that will be set apart from the telerobot for viewing the entire work area.

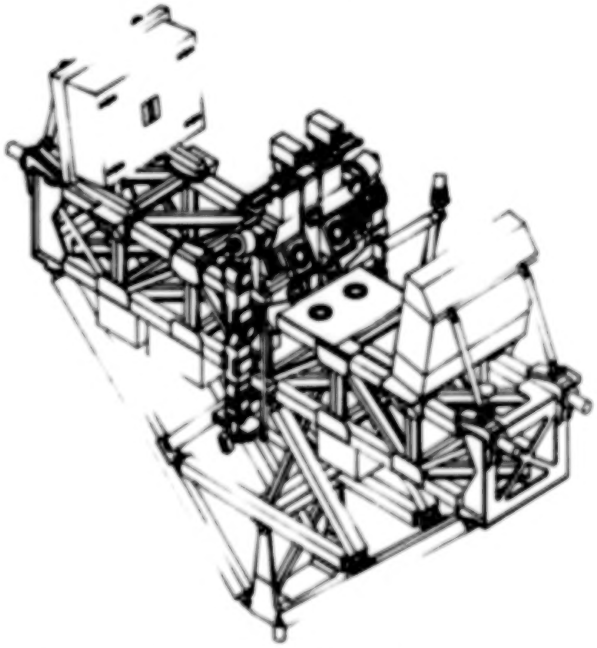


Figure 2. Configuration of Payload Bay Elements for Development Test Flight

In 1993, the DTF-2 will be flown with a mature version of the SSFTS to verify and demonstrate the system's capabilities to perform representative tasks for which the FTS will be responsible as part of its assembly, maintenance, servicing, and inspection tasks on Freedom Station. The workstation will be located on the aft flight deck of the orbiter, as in DTF-1, but this workstation will be the final SSFTS shuttle workstation which will be flown during the assembly flights of S.S. Freedom. DTF-2 will use the shuttle Remote Manipulator System (RMS) for mobility in the payload bay. The RMS will attach its end effector to a power and data grapple fixture located on the back of the telerobot and transport the telerobot to a worksite. These manuevers will involve close coordination between the operators of the two manipulator devices and will demonstrate the cooperative capabilities of the total system. The RMS provides the large motion and large torque capabilities, while the FTS provides the fine motion and dexterous manipulation. The FTS workstation will be located to the left of the RMS operator's workststion so that close communications can be maintained. Each operator will have an aft flight deck window located above his workstation for occasional viewing of the payload bay activity.

Evolution

The FTS system is designed for growth and evolution over the years. Initially it will be a teleoperated device with some autonomous functions for accomplishing routine operations in a well-structured environment. Over the 30-year lifetime of Freedom Station, the FTS will be upgraded through software and hardware changes to higher levels of autonomous activity. The vision system, which starts out as a closed-circuit television system, will expand to stereo vision, including some autonomous image recognition and location capabilities.

This capability for growth is only possible through careful planning and systems engineering at the outset of the program. A functional architecture was selected early in the program to guide the development of the software and hardware. This is the NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM) [4]. The software and computer architectures support this functional architecture so that orderly growth can be accomplished without major hardware changes. The SSFTS subsystems are designed with capacity margins for growth.

TECHNICAL OVERVIEW

System Description

The FTS system consists of a telerobot capable of working on a space station, shuttle, or OMV, a workstation for the shuttle orbiter, a workstation for the space station, spare parts and the storage accommodation equipment for storing the telerobot and its spares on the space station. Included as part of the telerobot are two manipulator "arms," an attachment, stabilizing, and positioning subsystem (ASPS) or "leg," cameras and lights, and all end-of-arm tooling.

The workstations are the principal point of human interaction for the control of the telerobot. Each workstation will be equipped with two six-degree-of-freedom (DOF) minimaster force-reflecting hand controllers. Video displays in the workstation will be capable of displaying up to three video images simultaneously, or two video images with health and status information displayed on the third screen. The telerobot cameras will be voice controlled, allowing the operator to maintain both hands engaged in manipulator teleoperation.

The telerobot will have a set of tools and end effectors that can be autonomously selected through the use of the end effector changeout mechanism (EECM) located on the end of each manipulator. Tool holsters located on the front of the telerobot body store extra end-of-arm tooling when not in use.

There are three ways that the FTS can be operated: fixed-base dependent operation, fixed-base independent operation, and transporter-attached operation.

In the fixed-base dependent operation, the telerobot is attached and stabilized to a worksite and derives its power, data, and video from an integral connection at the worksite attachment or an umbilical to a nearby utility port.

In the fixed-base independent operation, the telerobot is attached and stabilized to a worksite but derives its power from its own internal batteries. Communication between the

telerobot and the workstation is by a wireless link through the space station communications system.

In the transporter-attached operation, the telerobot is attached to an external transporter device, such as the shuttle RMS, the space station Mobile Servicing System (MSS) with the SSRMS, or the OMV. The telerobot derives its power, data, and video by way of a hardwire connection from the host transporter.

The total weight of the telerobot and the shuttle workstation will not be more than 1500 lb. In the stowed configuration, the telerobot will fit in a volume that is 7 ft x 3.5 ft $\times$ 3 ft. The power required by the SSFTS will not exceed 2000 W peak power, 1000 W average power, and 350 W standby power.

The SSFTS will have a system accuracy of less than 1.0 in. in position and $\pm 3.0°$ in orientation. System accuracy is defined as the ability to position and orient the tool plate center within a sphere of a given radius and a given tolerance in roll, pitch, and yaw relative to a Cartesian coordinate frame with the origin at the worksite attachment interface.

Space Station Freedom Interfaces

The SSFTS is designed to help in the assembly and maintenance of the Space Station Freedom and as such needs to operate off both the Space Transportation System (STS) shuttle orbiter and S.S. Freedom. Operations off of the STS will occur during the early Freedom Station assembly flights, and the SSFTS will make use of the standard STS interfaces, with perhaps the addition of a special umbilical for RMS operations. The SSFTS will require a workstation in the STS aft flight deck and interfaces to the STS power, data, and video systems in the payload bay. The RMS will be employed to move and position the SSFTS for operations on the STS. This requires the SSFTS to be equipped with an RMS-compatible grapple fixture for structural attachment to the RMS. As mentioned previously, an umbilical may be needed to meet the SSFTS power, data, and video requirements when operating from the RMS.

Once the SSF early assembly flights are completed, the SSFTS will begin operating off Freedom Station. This requires SSFTS workstation interfaces with the SSF Multipurpose Applications Console (MPAC) and interfaces to the SSF Data Management System (DMS), Operations Management System (OMS), Communications and Tracking (C&T) and Electrical Power System (EPS).

The MPAC will serve as the SSFTS Workstation with the addition of SSFTS-unique components. Currently, the FTS project intends to provide two six-DOF force-reflecting hand controllers, two standard data processors, and an adjustable restraint system to augment the MPAC. The MPAC is also the SSFTS interface to the OMS.

Other SSFTS interfaces fall into the areas of transportation, resources and storage. The SSFTS is designed to be transported by and operated from the Space Station Remote Manipulator System (SSRMS) and will meet the SSRMS mechanical, power, data, and video interfaces. The same grapple fixture on the SSFTS used for attachment to the RMS will be employed for attachment to the SSRMS. The SSFTS is compatible with the Freedom Station 120-Vdc EPS, Ku-band C&T, and DMS systems. Operations away for the SSRMS are possible in two basic modes. The first is the fixed-base dependent mode and requires a worksite attachment fixture that provides mechanical, power and data interfaces to the SSFTS. The second mode is the fixed-base independent mode and only requires a mechanical interface for attachment and stabilization. In this mode, the SSFTS receives power from its own battery and data through the S.S. Freedom C&T system.

Additionally, the SSFTS requires storage accommodation on the SSF. This includes power for battery charging and checkout, data for health and status monitoring, mechanical attachment, and storage for associated equipment and spares.

The SSFTS is also designed for Intravehicular Activity (IVA) maintenance. This limits the maximum size of the SSFTS so that it can be passed through a space station hatch.

SUBSYSTEM DESCRIPTIONS

Manipulators and End-of-Arm Tooling

The FTS telerobot contains a pair of 7-DOF manipulators, approximately 5 ft long from the shoulder to the tool plate. The kinematics of the manipulators are symmetric with roll-yaw-pitch at the shoulder, pitch at the elbow, and pitch-yaw-roll at the wrist. The manipulators are capable of producing a tip force of 20 lb anywhere within the work envelope.

The manipulator joint actuators each include a brushless dc torque motor, harmonic drive transmission, output torque sensor, output position sensor, fail-safe brake, cable wrap, and the housing and bearings required to carry structural loads. The brakes are designed to release when power is applied and engage when power is removed. The three shoulder actuators are all of common design and each produces 99 ft-lb of torque. The elbow actuator produces 58 ft-lb of torque. Each of the three wrist actuators, which are of similar design, produces 24 ft-lb of torque.

The manipulators have a repeatability of less than 0.005 in. in position and $\pm 0.05°$ in orientation. The incremental motion of the manipulators is less than 0.001 in. and less than 0.01° at the center of the tool plate.

The manipulators are backdriveable to allow stowing by an astronaut on Extravehicular Activity (EVA) or by another manipulator. The manipulators have camera assemblies mounted on the wrist roll assembly to allow the operator a close view of the end effectors and tools and the objects that they are manipulating. A force/torque sensor is mounted on the end of the manipulator to measure the forces and torques produced at the tool plate. The tool plate accommodates passthrough of power, data, and video to the end effectors and tools. The tool plate also accommodates the manipulator affixed element of the end effector changeout mechanism (EECM) by which tools and end effectors are automatically exchanged by the telerobot.

Attachment, Stabilizing, and Positioning Subsystem

A third appendage or leg on the telerobot is the ASPS. Its primary purpose is to attach the telerobot to the worksite and to position the body so that the manipulators can properly

approach the task. The ASPS must be stiffer than the manipulators and be capable of locking rigidly in place so that the forces and torques generated by the end effectors and tools can be properly transferred to the attachment mechanism.

The ASPS is a five-DOF manipulator, a little over 4 ft long from its base to the tool plate. The kinematics are roll-pitch-pitch-pitch-roll. The actuators are each capable of 24 ft-lb of torque. The torque capabilities when the brakes are locked is 180 ft-lb in the two shoulder actuators, 210 ft-lb in the elbow, and 240 ft-lb in the two wrist actuators. The minimum braked stiffness is 200,000 ft-lb/radian.

On the end of the ASPS is the Worksite Attachment Mechanism (WAM) by which the telerobot attaches to a fixture located at each worksite [see Figure 3]. For fixed-base dependent operations, the WAM makes electrical connection to the space station power, data, and video systems at the same time that it makes the mechanical connection. Self-aligning scoop-proof connectors in the WAM are mated when the WAM pulls itself into the attachment fixture.

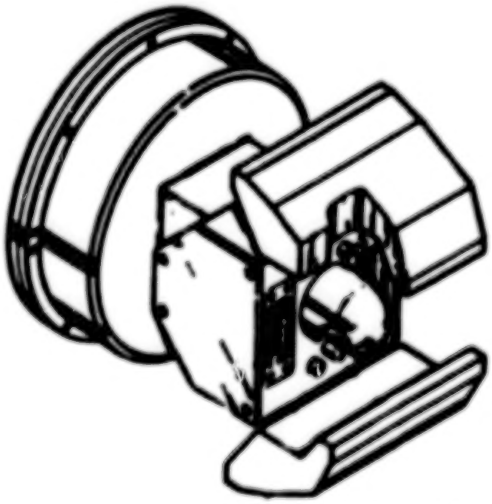


Figure 3. Worksite Attachment Mechanism

Data Management and Processing Subsystem

The Data Management and Processing Subsystem (DMPS) hardware and the software that executes in it are critical subsystems of the FTS. The SSFTS DMPS is a highly distributed processing system comprising multiple computers and networks which meet the stringent S.S. Freedom safety and reliability requirements. The DMPS implements a fault tolerant, redundant architecture which provides extensive growth capability because of its modularity.

The SSFTS DMPS is physically distributed: throughout the robot itself, in the FTS storage accommodation equipment, and packaged with the hand controllers. In addition to the computers provided by the FTS, the DMPS interfaces with the computers contained in the space station MPAC which is used as the work station by the astronaut FTS operator.

The FTS approach toward data management and processing is based on commonality with the space station. All FTS processors are in the 80386-80387 family of computers, which is the S.S. Freedom standard. The computers are connected through standard space station networks. All FTS processors access 1553b networks.

The computers internal to the robot use redundant 1553bs for communication. The SSFTS computers housed in the storage accommodation equipment are also connected to the space station Fiber Distributed Data Interface (FDDI) network.

The FTS computers include both space station Standard Data Processors (SDPs) and special purpose FTS joint controllers. The SDPs are used for high-level control and monitoring functions. The joint controllers perform servo level control of the manipulators, end-effectors, cameras, hand controllers, etc. Redundant processors provide backup capability in the event of failures of primary processors.

The SSFTS DMPS provides approximately 40 Million Instructions Per Second (MIPS) of processing power distributed among 18 joint controllers and 4 SDPs. The robot contains 14 controllers and 2 SDPs. The storage accommodation equipment houses 2 SDPs. The portable hand controllers contain two joint controllers. Later, additional capacity will be added as the FTS evolves toward autonomy. As an example, additional computers could be added for vision processing.

The flight software that executes in the DMPS must perform complex real-time processing. It is both CPU and input/output intensive. The basic control cycle runs at 50 Hz, and so the teleoperation of the manipulators appears instantaneous to the operator.

Every 20 ms the software must complete all processing and communications associated with the control cycle. This includes sampling the sensors in the hand controllers, interpreting the sensor data as either position or resolved rate commands, integrating the commands into Cartesian coordinates, converting the Cartesian commands through closed-loop form inverse kinematics into joint angle commands for the manipulators, and sending the commands to each actuator in each manipulator and end effector. To close the loop, the entire sequence is reversed by sampling the sensors in the robot, translating coordinates, scaling and indexing, and commanding the actuators in the hand controllers to provide force feedback. In addition to the basic control cycle, the FTS flight software must monitor and control all FTS subsystems, perform collision avoidance processing, support all the text, data and graphics interaction with the operator, and interface with the space station OMS.

NASREM provides the architecture for the FTS flight software. NASREM defines a set of standard hierarchical and horizontal modules and interfaces that correspond to different levels of autonomy. By adhering to NASREM, the FTS software can be developed incrementally. Initially, FTS will be primarily a teleoperated machine. With time, increased capability will be added allowing for the evolution to robotics. By enforcing the NASREM architecture on the FTS software, the addition of new modules and the exchange of existing modules with better algorithms is facilitated. NASREM will be used for all FTS flight software, including the two shuttle test flights. By SSFTS, the flight software will implement the first four levels of NASREM.

The FTS flight software will be implemented in Ada. It will be developed with the NASA Information System Life Cycle

methodology. It will be rigorously validated prior to each mission in a high-fidelity software test bed.

Workstation and Hand Controller Subsystems

The workstation is the man-machine interface to the FTS, providing the displays and controls that permit the FTS to be operated by a single individual. The degree of human interaction with the workstation and its location is a function of the evolutionary state of the FTS. Initially, FTS requires teleoperation through master hand controllers that are located on orbit with the slave manipulators. Teleoperation from the ground is impractical because of the time delays associated with RF communications. In the future, when the FTS evolves into an autonomous robot, hand controllers will no longer be the primary means of control. At that time, the FTS workstation may be on Earth.

During the two DTF flights and early S.S. Freedom assembly missions, the FTS will be operated by an astronaut from the shuttle Aft Flight Deck (AFD). Later, when the Freedom Station is more complete, the SSFTS will be operated from workstations located inside the pressurized modules. When the autonomous FTS is mounted on an OMV, it will be operated from a workstation that could be located anywhere, perhaps even on the ground.

The FTS STS workstation is provided by the FTS contractor. It consists of a display assembly panel, hand controllers, restraint system, and electronics. The hand controllers and the display assembly panel are stowed in the middeck lockers for STS launch and landing. The electronics are mounted before launch in the L10 payload station. The FTS display assembly panel will be unfolded by the astronaut and mounted to the A8 panel prior to FTS activation. It contains three color flat panel displays that will be used by the operator for viewing of any three FTS video cameras. Optionally, one of the displays can be used by the operator to monitor computer generated text and graphics. The FTS shuttle workstation also contains audio caution and warning indicators and lights, voice recognition hardware, keyboard, various hard-wired control switches, video recorders, data recorders, and assorted programmable function keys. The STS FTS restraint system, which is required because of the torques caused by the hand controller force reflection, consists of the Mission Specialist's chair reversed to face aft. The FTS hand controllers are mounted on the chair by the astronaut on orbit.

The SSFTS workstation, depicted in Figure 4, will use the standard MPAC, which provides the electronic interfaces between Space Station Freedom application programs and onboard operators. This will be augmented by portable FTS hand controllers, so that FTS can be operated from one of several workstations. The restraint system for the SSFTS workstation will be designed on the basis of the evaluations of operator performance during the shuttle test flights.

The FTS hand controllers are Martin Marietta/Kraft six-DOF, force-reflecting hand controllers, as illustrated in Figure 5. The FTS design is predicated on a mature design that has been in use in nuclear and undersea applications since 1980 and is the leading force-reflecting hand controller in use today. Among the improvements for FTS are shifting the wrist joints to a coincident point of rotation and adding force feedback to the wrist

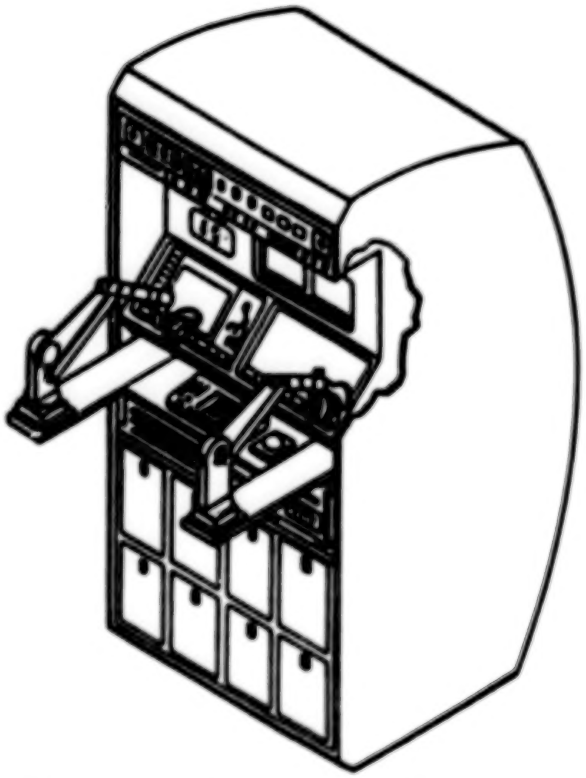


Figure 4. Space Station Freedom FTS Workstation

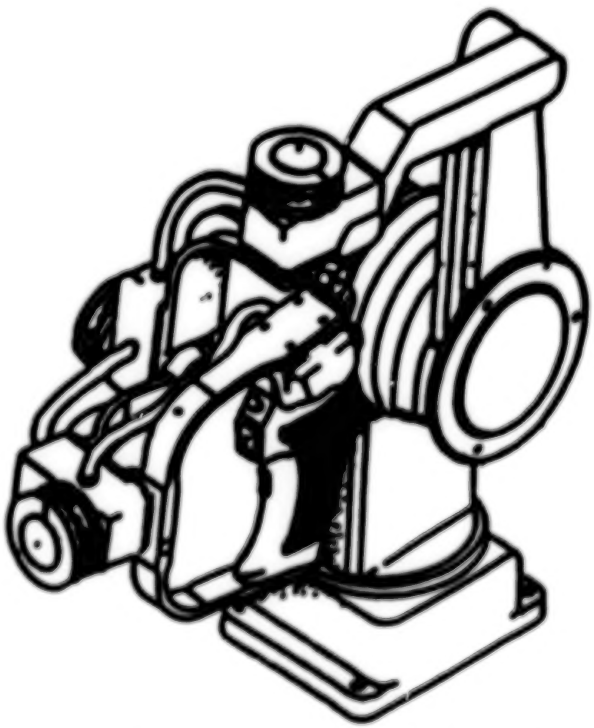


Figure 5. Six-Degree-of-Freedom Hand Controller

roll. The hand controller developed and tested on DTF-1 will be the basic approach for all FTS missions.

The FTS hand controllers are easy to use, because they support an intuitive relationship between the operator's shoulder, elbow, and wrist movements and those translated to the manipulator. They can be used for both position and rate control. Because of the force reflection requirements, the FTS hand controllers themselves are robots having actuators and sensors, and they are tightly coupled to the FTS control system. The FTS operator will use voice commands to control the cameras, so that he does not have to let go of the hand controllers while he is working. Switches on the hand controllers are used to enable any potentially hazardous activity, such as changing end effectors.

Vision Subsystem

The SSFTS Vision Subsystem consists of color cameras, lighting, and video switches. There is one camera on each manipulator wrist, with positioning provided by the manipulator. Two head cameras are mounted on Camera Positioning Assemblies (CPA) for global viewing. Also, the ASPS is scarred for the later addition of a camera for use in autonomous docking. Focus, aperture, head camera positioning, and light are all controlled by the vision subsystem. All cameras are color, using charged coupled devices (CCD). Each light will have the capability of providing up to 100 foot-candles of illumination. The four video channels are routed through a 6 by 4 video switch to either analog-to-digital (A/D) converters for the radio frequency (RF) link or an umbilical for transmission to the SSF C&T video processors. The C&T video processors select and process the signals to be displayed on the workstation monitors. For the early flights, when the SSFTS is operating from STS, the four channels will be transmitted via umbilical to a 5 by 3 video switch in the payload bay. Three of the four channels can be selected for display at the STS workstation.

Communications Subsystem

The Communications Subsystem (COMS) provides the two-way RF communication link between the telerobot and the workstation, and the one-way RF EVA safing link between the EVA astronaut and the telerobot. The communication links provide command data from the workstation to the telerobot, and telemetry data, and video from the telerobot to the workstation.

The COMS consists of a Ku-band Video/Telemetry/Command data transceiver and the EVA safety shutdown functions of a Ku-band EVA safety shutdown transmitter and EVA safety receiver.

The COMS is a Ku-band transceiver and is compatible with the SSF C&T system. The COMS modulates and transmits telerobot video data (up to four video channels) and telerobot health and status data on two Ku-band carriers. These carriers are received and processed by the SSF C&T. Likewise, the COMS receives and demodulates telerobot command and control data that is transmitted from the space station C&T.

The COMS also receives power-up command and control, and transmits status and health when queried, from within the telerobot storage accommodations on space station.

The COMS module houses the baseband modulator microwave transmitter and command demodulator and provides all necessary interfacing to the computers and power system. The COMS amplifies and transmits the digitized video and telemetry data on two carriers at 14.63 and 14.67 GHz, respectively. The EVA safety receiver also operates at Ku-band, but its operating frequency has not been determined.

The antenna assembly consists of one Ku-band circularly polarized omnidirectional antenna. Two antenna assemblies are mounted on the telerobot, one located on the top and the other on the bottom of the telerobot. The Ku-band antenna is approximately 0.7 in. in diameter and 0.5 in. high and sits on an extendable boom that can be retracted for telerobot stowage.

The EVA safety shutdown transmitter is on the EVA suit. Each EVA safety shutdown transmitter will be activated by a simple switch. To satisfy the fail-safe requirement, the transmitter will transmit a "heartbeat" version of the safety shutdown code when safety shutdown is not activated to indicate a healthy transmitter.

Power and Electrical Subsystems

The Power/Electrical subsystem receives power from the SSF or NSTS and provides conversion, regulation, and distribution of power for telerobot use. An independent power capability is provided by an internal battery for detached telerobot operation, to provide uninterruptible power to safety critical loads and to maintain keep-alive power for critical telerobot memory. Load control and circuit protection for FTS loads and interfaces are also provided by the Power/Electrical Subsystem.

The Power/Electrical Subsystem interfaces with NSTS power at 28 Vdc and SSF power at 120 Vdc. The 120-Vdc power from the SSF is conditioned (with a 120- to 28-Vdc converter) to provide a common 28-Vdc "Main Bus" voltage for distribution at 22 to 32 Vdc within the telerobot. The Battery Module Unit consists of three NASA standard 20 Ah batteries and a dedicated charger. The batteries are sized to support 2 hr of telerobot detached operations.

Thermal Control Subsystem

The SSFTS is required to operate under all environments for indefinite task durations while working on the NSTS and SSF. The SSFTS thermal design meets this requirement with an approach that is fundamentally passive, relying on selected coatings, special shielding, and carefully chosen equipment placements and mountings. It is augmented with controlled electrical heaters on selected components to compensate for varying power dissipation levels.

The exterior surfaces of the SSFTS are used to balance external heat loads against heat loss to space and thus maintain required temperature levels and to reduce sensitivity to orbital and orientation environmental variations. Internally mounted equipment boxes and interior surfaces are generally coated with a flat absorber (black) type coating for interior group component temperature control.

Workstation equipment are located in the pressurized compartments with the crew. This equipment is maintained within their

allowable temperature limits by air cooling in the STS aft flight deck or Freedom Station node.

Control Algorithms

The FTS control algorithms support all the telerobot operations in both teleoperated and autonomous modes. They also support bilateral force reflection, which enables the operator to experience the forces and torques sensed by the force/torque sensor at the tool plate of the manipulators. Bilateral force reflection has a number of advantages for teleoperation. It improves safety by giving the operator immediate confirmation that the manipulator has come in contact with another object. It reduces training time and eliminates errors by giving the operator a more natural feel for the operations and the environment.

But force reflection comes with a price, which is the low data latency requirement which must be satisfied by the data system on the space station. For force reflection to be useful, it requires a minimum around-the-loop control rate of 50 Hz. This means that all the control computations and the data transfer from the hand controllers to the manipulators and from the manipulators back to the hand controllers must be accomplished in 20 ms. Half of this 20 ms is allocated to data transmission, which means that the control calculations must be completed in 10 ms under all operating conditions.

Initial tests using coded algorithms in machines of equivalent speed to the flight computers indicates that the 10 ms is achievable with some margin. The biggest uncertainty is whether the space station Data Management System (DMS) will be capable of meeting the FTS data latency requirement, considering the amount of traffic on the DMS bus. Consideration is being given to a dedicated bus for the FTS in the event that the common bus cannot meet the requirement. Such action has been taken with the space station Guidance, Navigation, and Control System, which also has stringent time delay requirements.

The control algorithms provide a number of features that make the FTS a safe and useful tool for the astronauts. The operator will have the capability of selecting and defining coordinate reference frames, and he will be able to perform dual-arm coordinated control of a grasped object with a single hand controller. The control algorithms are also capable of shared control in which the operator controls motion in one or more coordinate axes, and the telerobot autonomously controls the motion in the other axes. The algorithms provide a smooth, safe transfer between autonomous and teleoperation control. They also ensure that manipulator singularities can be driven through and that joint stops can be reached and recovered from. There are backup methods of control being investigated, so that the operator will be able to reconfigure the FTS for safe transport in the event of a failure of the primary system.

OPERATIONS SUPPORT

Although the hands-on operation of the FTS will be by the astronauts on orbit, there are a number of activities that will be conducted on the ground in support of the on-orbit operations. A number of these operations look to the future by developing new task scenarios as new jobs are identified for the FTS and by developing new hardware and software for the future growth and evolution of the FTS. Still other operations are involved in the logistics and maintenance functions necessary to sustain a system for the duration of the Freedom Station design life of 30 years. Finally, there are the normal engineering ground operations that involve the collection and monitoring of data and the generation of the appropriate commands and software loads. A number of facilities, equipment and approaches are being developed to accomplish these activities.

Task Analysis

The GSFC Mission Utilization Team (MUT) has evaluated potential FTS assembly, maintenance, and servicing tasks through the use of the Task Analysis Methodology Document developed by the team [5]. The output for each task is an operational script that isolates individual task activities of the FTS, the RMS, the Astronaut Positioning System (APS) and the Mobile Servicing Centre (MSC). In addition, the stability/resource attachment points for the FTS ASPS and handhold locations are defined. Reach capabilities are assessed either through use of Computer Aided Design (CAD) video models or small-scale physical models with the proper kinematics.

This process has identified tasks that contribute to the assembly of Space Station Freedom and reduce the amount of crew EVA time. One of the potential tasks currently under investigation is the installation of resource pallets by using the FTS. These are the large, tablelike structures that attach to the nodes of a truss bay and support elements of the various space station systems, such as the Guidance, Navigation, and Control System.

The operational approach is to perform FTS tasks before EVA during a 3-hr period when the astronauts are preparing for the EVA. The task analysis must be sensitive to the operational flow; e.g., a truss must be built prior to the installation of a pallet by the FTS. Initial studies have the FTS working independently from the astronauts.

Figure 6 shows the FTS attached to the APS during installation of a typical pallet. The RMS has located the pallet so that the four legs are in the vicinity of their respective attachment points at the truss nodes. The FTS softdocks the pallet to the truss while stabilizing with one manipulator system to a truss node. After performing the leg attachment, the APS moves the FTS to each leg in turn. For those pallets where the fourth attachment is outside the APS/FTS reach, the RMS releases the pallet (which is still held at three attach points), attaches to the FTS grapple, and transports the FTS to the final attachment location. The FTS operates while on the RMS for this operation and for the connection of the pallet utility harness to the space station utility tray. Local stabilization is achieved by attachment of one manipulator to the truss, pallet, or utility tray as required.

Facilities and Equipment

There will be a number of facilities at the GSFC and the Johnson Space Center (JSC) to support the on-orbit operations, maintenance, and growth of the FTS. Prior to any of the FTS missions extensive crew training will be conducted by the JSC at their facilities. A high-fidelity crew trainer will be installed at JSC for this activity. Also, a telerobot simulator will support crew training. This is a real-time, kinematic simulation of the telerobot. It will exercise all the workstation interfaces,

taking hand controller inputs and driving all the functions in the workstation except force reflection. System kinematic information will be available for display, including graphic representations of all telerobot camera views.

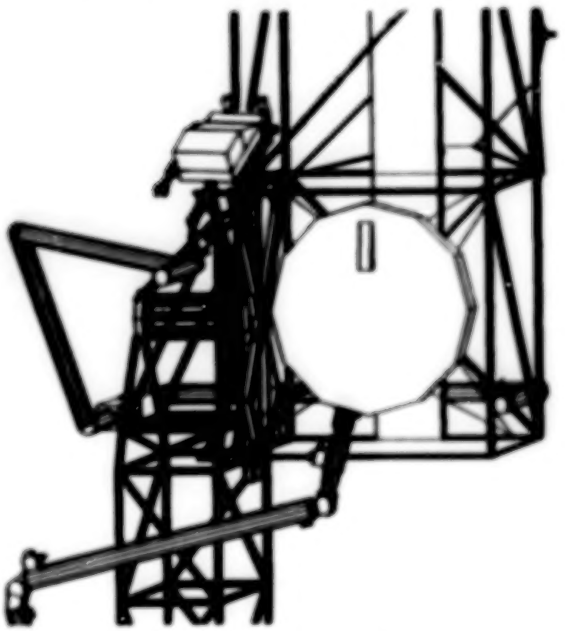


Figure 6. FTS Attached to Astronaut Positioning System

Full-scale kinematic mockups of the telerobot will be available for testing in the JSC Weightless Environment Test Facility (WETF). These will be passive mockups that will be used to verify EVA interfaces and handling, such as manual release of the attachment mechanism, restowing and transporting back through the space station hatch for repair. There will also be light-weight mockups for use on the JSC Manipulator Development Facility (MDF) to exercise the interfaces and operation with the RMS.

When DTF-2 is returned from its mission it will be refurbished to become a ground-based version of the FTS, called the Engineering Test System (ETS). The ETS is a functional duplicate of the SSFTS, the space station workstation, and the storage accommodations. The ETS will be housed at GSFC in the new Spacecraft Systems Development and Integration Facility (SSDIF). The ETS will serve as a high-fidelity testbed for the development and testing of flight hardware and software. It will also be used to support on-orbit anomaly investigation.

The primary facility for monitoring the FTS operations and for the analysis, archiving, and trending of data will be the Engineering Support Center (ESC) at GSFC. The FTS will share part of the Work Package 3 ESC and will provide its own workstations and support computers as needed. Commands and software loads will be generated in the ESC and transmitted to the space station control center (SSCC) at JSC for relay to FTS. Similarly the downlink data comes to the ESC through the SSCC.

There will be a software development facility (SDF) at GSFC for the maintenance of the flight software and for the generation of new code as new capabilities and new hardware come on line and as new task scenarios are developed.

The present GSFC robotics laboratory or Development, Integration, and Test Facility (DITFAC), as it is officially known, will support the FTS operations as an advanced development testbed. Candidate hardware for future addition to the FTS will be tested and evaluated in the DITFAC before being selected for implementation. Such activities as tool development and evaluation and task element design and test will be conducted here. Currently the laboratory is involved in a number of activities in support of the FTS development, including the evaluation of hand controllers, Orbital Replaceable Units (ORUs), task scenarios, camera locations, and workstation designs.

It is necessary that all of these ground support facilities be interconnected with voice, video, and data links so that the operations support can be properly coordinated and managed. Work is currently underway to determine the requirements for these communications links.

CONCLUSION

The FTS project is ready to start the detailed design and implementation phase, confident of the results of the phase B preliminary design work and the considerable effort by Martin Marietta in preparation for the DTF-1 preliminary design review. The FTS promises to be a safe, reliable, and useful tool for the astronauts on both the space shuttle and S.S. Freedom. The program's success is promoted by the cooperation we are receiving from the JSC for system interfaces, human-machine interfaces, and crew training, and the program office in Reston, Virginia, for the overall guidance, the allocation of resources and the establishment of stationwide interfaces.

REFERENCES

1. Andary, J.; S. Hinkal; and J. Watzin; "The Flight Telerobotic Servicer (FTS): A Focus for Automation and Robotics on the Space Station," Paper IAF-87-25, 38th Congress of the International Astronautical Federation, Brighton, U.K., October 10-17, 1987; Acta Astronautica, Vol. 17, No. 8, August 1988.

2. Andary, J.; S. Hinkal; and J. Watzin; "Design Concept for the Flight Telerobotic Servicer (FTS)," presented at the 2nd Space Operations Automation and Robotics Workshop (SOARB8), Dayton, OH, July 20-23, 1988; NASA Conference Publication CP-3019, pp. 391-396.

3. Final Report, Flight Telerobotic Servicer (FTS) Titman Concept In-House Phase B Study, Volumes I and II, SS-GSFC-0042, September 9, 1988.

4. Albus, J.; H. McCain; and R. Lumia; NASA/National Bureau of Standards (NBS) Standard Reference Model for Telerobot Control System Architecture (NASREM), SS-GSFC-0027, December 4, 1986.

5. Mission Utilization Team, Flight Telerobotic Servicer (FTS) Task Analysis Methodology (TAM), April 14, 1989.

SATELLITE SERVICING

C. Woolley and J. Moore
NASA/Johnson Space Center

(Paper not provided by publication date.)

THE TELEROBOT TESTBED: AN ARCHITECTURE FOR REMOTE SERVICING

J. R. Matijevic
California Institute of Technology, Jet Propulsion Laboratory
4800 Oak Grove Drive, M/S 303-308, Pasadena, California 91109

ABSTRACT

The NASA/OAST Telerobot Testbed will reach its next increment in development by the end of FY'89. The testbed will have the capability for: force reflection in teleoperation, shared control, traded control, operator designate and relative update. These five capabilities will be shown in a module release and exchange operation using mockups of ORUs. This development of the testbed shows examples of the technologies needed for remote servicing, particularly under conditions of delay in transmissions to the servicing site. In this paper the following topics are presented: the system architecture of the testbed which incorporates these telerobotic technologies for servicing, the implementation of the five capabilities and the operation with the ORU mockups.

INTRODUCTION

From its inception the NASA/OAST telerobot testbed project has been asked to investigate the technologies in the disciplines of telepresence, robotics and artifical intelligence in application to remote servicing. The technologies chosen for integration, system test and demonstration in the testbed have the potential to usefully complement, significantly enhance or even replace selected manned space activities through engineering development and qualification in flight telerobotic systems. Indicative of this potential, certain generic tasks, suggestive of space assembly, maintenance and repair, are performed in the testbed. Through performance in several modes: direct teleoperation with/without force reflection, shared control, traded control between teleoperation and the autonomous system, and robotic operation, the benefits of the individual technology contributions to the operation can be quantized and recommendations for use in telerobotic systems established. This paper reports on the development leading to the first such quantization, using an ORU removal/replacement task. Several recommendations are offered for near-term telerobot system developments, based on the experience of this testbed project.

ISSUES IN REMOTE SERVICING BY TELEROBOTS

In considering the use of telerobotic systems in remote servicing operations several factors will guide the application. These factors form an initial set of system requirements and are derived from the experience base of manned space servicing operations.

Telerobots must work in the same domain as astronauts in EVA

For the near future, the opportunities for remote servicing of spacecraft will be limited to those spacecraft designed for service by astronauts in EVA. Consequently, telerobots must be designed to work with the fixtures and devices of ORU's, fluid couplers, hand holds and tools for astronaut servicing. Options for robot friendly designs will be limited to such as: end-effector attachment accommodation for tools and hand holds, LED type markings for object tracking by camera/laser systems, and power and signal ports for attachment of the telerobot. Fixturing and jigging of the worksite, a common practice for industrial robots, will be restricted to new operations for the telerobot such as assembly.

Additional complexity for the telerobot will be afforded by the environment at the worksite for service. Experience in past servicing missions has shown this environment to be variably if not poorly lit (by present day robotic laboratory standards) with restriction on access or obstacle strewn.

Telerobots must be capable of adapting to changes in the workplan

The history of manned servicing missions (e.g., Solar Max, Syncom 3, see [4] also [7]) has shown that well planned and rehearsed servicing procedures must be adapted to accommodate the unexpected. The telerobot performing a servicing mission must exhibit some of the same flexibility of an astronaut in EVA to adapt to errors in modeled behavior and develop workaround procedures.

A human operator must be able to supervise telerobots in operation

The presence of a human supervisor for a telerobot is inherited from the history of space operations. In early applications of telerobots to servicing in which telerobots must work along side astronauts in EVA, such supervisors will have a special responsibility for safety at the worksite for both the astronaut and the serviced item. Another dimension is added to this role when the supervisor acts as the operator of a telerobot system. A human operator is an essential component of a telerobot system from the commanding of the movement of manipulator arms through hand controllers to the establishing of frames of reference for and initiation of robotic operations. The flexibility in telerobot systems for adapting to unexpected conditions and developing workaround procedures will be the special contribution of the operator.

As a consequence there will be requirements for telerobot systems to provide a rich set of data feedback sources to the operator remotely located from the worksite. These sources (e.g., cameras, ranging systems, force/torque sensors, additional lights) will be additions to that nominally provided for monitoring and recording the activities of astronaut servicing in EVA.

Telerobots must be capable of performing a variety of tasks at different worksites

Past servicing missions have required the performance of a number of tasks by the astronaut. Although a primary task may be the removal and replacement of an electronics or instrument module (e.g., Solar Max), access to the module required removal/cutting of thermal blankets, release of threaded fasteners, detachment of electrical connectors, movement of the module around wire bundles and alignment/checkout of the replacement module. Servicing at more than a single site (or in the case of Solar Max more than a single side of the satellite) has also been a requirement. For a telerobot this implies that the system must be capable of establishing a frame of reference, updating that reference during the operation and performing a number of different manipulations at any such site.

Telerobots must be capable of operating under conditions of delay in transmission

The earliest missions for a telerobot will be service based at the STS orbiter or the Space Station, where the human operator will be an astronaut in IVA. The full potential for the contribution of telerobots will be realized with the expansion of operations to servicing missions remote from manned vehicles. In these instances, transmissions to the human operator, in all likelihood located at an earth-bound station, will be relayed through communication satellites and thus subject to variable delay on the order of 2-6 seconds. Maintaining the rich set of feedback for human interaction with the telerobot under these conditions will be the challenge.

In the following paragraphs the approach in the telerobot testbed project to meeting these system requirements for a servicing telerobot is discussed.

TELEROBOT TESTBED CAPABILITIES

The approach adopted by the first applications of telerobots to meeting the requirements of remote servicing will use forms of direct teleoperation with forces/torques reflected to the operator. Of the many types of developments possible, both the servicing telerobot for the Space Station, the Flight Telerobot Servicer (see [2]), and the telerobot testbed have chosen to implement a Cartesian or task space based system. The telerobot testbed approach is discussed below.

Force reflection in teleoperation

In a task space based teleoperation system the operator controls the manipulators by providing six position/orientation commands for the location of the end effector through a six degree-of-freedom hand controller. The electronics of the system develop the individual manipulator joint level commands necessary to drive the end effector to assume the commanded position, resolving any configuration ambiguity as necessary. Manipulator path planning, collision avoidance, arm coordination and object manipulation are all performed by the operator in real time through the hand controllers. The feedback to the operator of the effect of his manipulation is provided by the kinesthetic of the hand controller back driven by the input from force/torque sensors. In the testbed system, these sensors are mounted on the wrists of the manipulator and measure the forces and torques encountered by the end effector in contract at the worksite. The six forces and torques so measured are transformed into the commands suitable to back drive each of the joints of the hand controller.

Given the task space reference for such a system, there is a natural way for the hand controller to be referenced so that manipulation by a tool can be performed in the same style as that by the end-effector. The hand controller can also have a smaller kinematic and dynamic range than the manipulator, affording a compact configuration of the operator workstation. These capabilities are provided by the functions of the system to accept a change to the coordinate frame of reference, to be indexed and scaled.

The performance of such a force reflecting teleoperation system has been recently characterized (see [3]) in terms of the latency in the round-trip passage of data in the system. By round-trip passage is meant the transmission of (a) data derived from the input to the hand controller, to (b) data which commands the joints of the manipulator to achieve a new position, to (c) data which is measured from the force/torque sensor, to (d) data which is used to back drive the hand controller. When the latency of this round-trip passage of data is under 10 ms the operator has an adequate 'feel' of the end effector in contact with the work site. Furthermore, as measured through time for manipulation, training and sum of forces applied during the task, the 10 ms threshold represents a clear performance boundary.

When because of delays in transmission or limitation in implementation the latency increases above this threshold, other forms of telerobotic operation should be considered for a servicing application. One such is shared control as discussed next.

Shared control

In a general sense, shared control allows for manipulator control to be jointly performed in real time by both an operator in teleoperation and an autonomous system. In the telerobot testbed two examples of such shared control are being developed. In the first such example, the location of an object grasped by two manipulator arms is controlled through the inputs from a single hand controller. The autonomous control of the telerobot modulates the forces applied by the manipulators to the object with the goal of minimizing the net force. This control is exercised in real time with the control which results in the repositioning of the object under control of the operator. In the second example, the location of the end effector is controlled by the input from a hand controller. When in contact with the environment, the forces encountered are measured by the force/torque sensor and the autonomous control reacts to these forces in a position accommodation scheme. During a particular task, the operator is responsible for maintaining the end effector in position at the worksite. The autonomous system prevents inadvertent or extreme forces from being applied through the position accommodation control.

It is this latter example of shared control which suggests the approach to conditions in which a telerobot in servicing experiences latency in data transmission which prevents effective force reflection in teleoperation. In a shared control mode the telerobot's autonomous system can modulate or otherwise control the amount of force/torque applied during a task. The risk of damage from binding, galling, gauging or other inadvertent force application to satellite equipment can be mitigated when the operator may otherwise be unable to sense the extent of forces introduced by the telerobot.

When the latency in data transmissions begins to reach 1-2 seconds video and other forms of operator feedback begin to degrade. The delay in tracking the position of the manipulator arms and end effectors can result in inadvertent collisions with equipment at the worksite. For gross motions of the manipulators, this problem may be handled through the use of preview displays. In such a system, the motion of the hand controllers drives a simulation display of the manipulators overlaying the video feedback. The operator gets a preview of the effect of the motion of the arms through this display and can correct for potential collisions with the worksite. This technique of preview display cannot account for conditions in which the arms are already in contact with the worksite, since simulations of contact forces and mechanical compliance by the arms will be prone to error given reliance on geometric and dynamic models of the arms and the workpieces. Other types of telerobotic operations should be considered under conditions of this size latency. One approach taken by the telerobot testbed is a form of traded control discussed next.

Traded control

In the most general sense, traded control is the transfer of control between the operator in teleoperation and the autonomous control capability of the telerobot. In the telerobot testbed, the operator performs all gross motion planning and maneuvering of the end effector to the vicinity of a task at the worksite. The operator then transfers control to the autonomous system for motion of the end effector to contact with the worksite and subsequent performance of work. During this period the operator commands actions of the telerobot in a supervisory mode, maintaining the capability to intervene at any time to correct and/or stop the manipulation. When the work is completed, as judged by the operator, the system moves the arm/end effector away from the worksite and offers to transfer control to the operator for subsequent teleoperation.

Although this style of traded control is intended as a form of telerobotic operation which accommodates 1-2 second latency in transmission, this type of operation can be used effectively in conditions of restricted visual feedback to the operator. Autonomous alignments, contacts, seatings of end effectors and tools can be used to augment a basically teleoperated manipulation through forms of traded control. Traded control can work so long as the autonomous manipulations performed prove successful. Given reliance of this control on the models of the task, worksite and manipulators, techniques for accommodating errors in these models must be present to ensure success in the operations. The following two paragraphs present such techniques developed in the telerobot testbed.

Operator designate

For subsequent manipulation the operator can register the location and otherwise identify an object using the function of operator designate. In the telerobot testbed this function is implemented using a wire frame model of the object in an overlay of the video feedback from the worksite. The operator assigns, in a one-by-one fashion, the vertices of the model to the vertices of the object in the video. When sufficient vertices have been assigned, the system calculates the size and position of the object in the camera view using the models of the camera providing the image. By performing the operation in more than one camera view, (assuming sufficient separation of the cameras) a six dimensional vector giving the location of the object can be developed. The location and name of the object are registered in the autonomous systems data base for future manipulation. By a similar technique, regions of space at the worksite can be designated as 'forbidden' regions, thus enhancing avoidance of inadvertent collisions. Parts of the telerobot can also be designated, allowing for combined registration of the camera and arm models.

Since the object and telerobot will be moved during servicing operations the operator designate function allows the operator and thence the autonomous system to track the present locations of objects. This function does rely upon good camera models since the object so designated is registered in an absolute sense at the worksite. In the next paragraph this restriction is relaxed.

Relative update.

In a relative update to the data base of the autonomous system, the operator locates two objects at the worksite and calculates the distance between the objects. When the two objects are a workpiece and the end effector of a manipulator arm, the distance between the objects can be used to command the arm to contact with the workpiece. In the first application of this technique in the telerobot testbed, the operator designates the two objects. In a variation, a machine vision function can be used to provide the locations of the objects.

Since the calculation of the distance between the objects is performed essentially in the image buffer of the video, a precise camera model is not needed. The resolution is provided to the pixel level. By calculating in several views, a six dimensional frame vector can be developed affording the information necessary for grasps and other manipulations.

These five capabilities of the telerobot testbed yield an approach to the implementation of a telerobot for remote servicing. In the following the architecture which is an implementation of these capabilities is presented.

TESTBED ARCHITECTURE

Conceptually, the telerobot testbed architecture follows a hierarchical design philosophy. In this design the human operator and machine intelligent functions of the system are placed at the top of the architecture with the primitive or mechanical functions toward the bottom (see Figure 1). Five subsystems comprise the testbed system. A brief description of the functions each follows (see also [1])

Operator Control Station (OCS)

The OCS provides an efficient user friendly physical interface between the operator of the telerobot and the testbed system. An interface for two operators is in fact provided. For the operator of the telerobot a stereo display, a pair of 6 degree-of-freedom force reflecting hand controllers, graphical displays of the force/torque sensor measurements and monitors to the computer system of the OCS and the TPR (see below). A voice recognition system allows the operator to control the telerobot autonomous functions while controlling the telerobot manipulators through the hand controllers. An enunciator provides verification of input commands and status of progress of the system in operation.

A pair of video displays, a monitor to all telerobot subsystems, and a set of function switches for control of the OCS are provided for the second operator, the test conductor. This operator monitors the performance of the telerobot operator and the telerobot system during operation and sets up test conditions.

At the OCS the operator initiates all functions of the testbed. In particular, the function of operator designate is performed using the video displays at the OCS.

Task Planning and Reasoning (TPR)

The TPR performs functions of high level task planning and gross motion planning. In supporting the five capabilities of the testbed a menu for operator commanding of the autonomous functions of the system is provided. A kinematic simulator can be used to preview the effect of an operator initiated command.

The TPR also provides a data base of objects in the worksite, including their locations, connectivity and semantic relationships. During the testbed operations of operator designate and relative update, the TPR interacts with the OCS to update the data base with the locations of objects at the worksite. During autonomous operations in a trade of control, the data generated by the functioning of the sensing and manipulation capabilities of the system is assembled by the RTC for update of the TPR data base. When applicable, the data associated with an update is processed for spatial and semantic consistency.

Run-Time Control (RTC)

The RTC provides the fine motion planning and execution of autonomous operations by the telerobot. The RTC sets up the sensing and manipulation functions in the testbed for autonomous operation or teleoperation. RTC monitors execution and interacts through the TPR with the operator to direct recovery in case of failure.

As a part of the fine motion planning, the RTC contains a kinematics simulator which plans motions of the manipulators to avoid joint stops and singularities. A run-time data base of the locations of objects at the worksite is updated during operations and is used to plan for detection and avoidance of collisions. Knowledge of the kinematic, dynamic and inertial properties of the manipulator arms and objects to be manipulated is kept as part of this data base.

Sensing and Perception (S&P)

The S&P contains the cameras and the machine vision functions of the testbed. These functions include a tracker for use in supply position/orientation of objects in the worksite. This tracker employs a model matching technique like that of the operator designate function using the images of an object in several views to derive a six vector of location. Although developed for tracking moving objects at the worksite, in supporting the five capabilities of the testbed, the tracker is used to determine locations of stationary objects in the worksite. The data from operator designate is used as an initial estimate of the location of the object in this function. The tracker performs a refinement on the location of the object, employing statistical correlation techniques to more precisely match model to image of the object. The data so developed can be used both in an absolute or relative mode for a subsequent manipulation.

Manipulation and Control Mechanization (MCM)

The MCM provides the manipulation capability of the testbed. This subsystem consists of three 6 degree-of-freedom arms, one of which serves as a camera platform, two servo controlled end effectors, the electronics and computing system which supports the data rates necessary for force reflection in teleoperation and both the autonomous and shared control capabilities of the telerobot. These latter functions of the MCM are provided through a set of macros, software routines instantiated by the data provided either by the autonomous system or the operator to perform certain manipulation skills.

One of the requirements for the development of the testbed system is the ability to accommodate growth. This is provided in part through the loose confederation of computing systems which comprises the testbed (see Figure 2). The use of a network for the basic intercommunication architecture allows easy expansion through the addition of computing resources. In addition, the existing computing systems can be arranged to form different telerobot configurations. One such configuration will be formed with the testbed accommodation of time delay. In one example, the OCS/TPR will be considered the local site and the RTC/S&P/MCM considered the remote site. With the buffering of commands from the OCS/TPR along the network and through the teleoperation system, a sense of latency in data transmission can be given to the operator.

Finally, the demonstration of the five capabilities of the telerobot testbed will be provided through an ORU removal/replacement task (see Figure 3). In this task, the site for placement of an ORU mockup (called the truss structure of the stow bin) will be moved to some location within the reach envelope of the manipulators. The operator will attempt to remove the module from a platform between the arms and insert it into the truss structure. The operator will be asked to first do the task in force reflecting teleoperation. Then the operator will perform the task using the functions of operator designate (locate the ORU and the place for insertion), relative update (determine the distance between the end effector of the arm and a grapple lug on the module, or the distance between the module and the place for insertion), and traded control (the operator moves an arm near the grapple location and trades to the autonomous system to grasp, or the operator moves the arm near the insertion point and trades to the autonomous system to perform the insertion). Lastly, the operator will be asked to repeat the operation using shared control. Each of these manipulations will be repeated under condition of 2 second latency in data transmission. In a separate demonstration the dual arm manipulation capability of the testbed will be shown using another ORU mockup.

DIRECTIONS FOR DEVELOPMENT

During the four year development of the telerobot testbed the experience of integration has led to certain choices in the utilization of the technologies in telerobotics based on the relative maturity of the technology for near term application and suggested by the environment for remote servicing. These choices are described below in the form of recommendations for development of a telerobot to perform servicing in a quasi-static environment (i.e., at a worksite where change in configurations is only introduced by the telerobot).

Human vision through operator designate can replace machine vision

In the earliest applications of the machine vision function of fixture verification (i.e., refining the knowledge of the location of a fixed object in the worksite) in the testbed, accuracies on the order of 2mm in position and 0.5 deg in orientation were achieved in locating an object in the worksite from a camera located within 1m of the object. Such update of the location of an object was required to ensure a successful grasp or other manipulation where clearances between gripper finger and object of +/-2.5mm were common. In order for the function to work, an accurate camera model and an initial acquisition of the object was required to allow the machine vision algorithms to converge reliably on the desired object. In the testbed the acquisition was provided through an a'priori generated data base which, with calibration, was accurate to within 5mm of the correct position. The accurate camera model was provided through a careful calibration of the camera platform (a PUMA arm) and camera.

Given the difficulties of obtaining much less maintaining a well calibrated model of the worksite or of the cameras in a remote servicing application, other approaches were considered for this latest integration of the testbed. The operator designate function with accuracies in locating the position of an object to 1cm from a distance of 3-4m would be sufficient for reaching the acquistion range of a fixture verification function. The relative update function with accuracies of +/-3mm from a distance of 1m would be sufficient to substitute for the data from the fixture verification function. In the latest testbed integration, this latter statement is particularly true since clearance requirements between gripper finger and object are at least +/-1cm. It should be noted that several centimeters of clearance are nominally required for astronaut in EVA grasp or manipulation of servicable items [6].

Relative update can compensate for uncalibrated camera views

In supporting the machine vision fixture verification function, the requirement for a calibrated camera model involved both calibration of the camera platform (a PUMA arm) and camera. Such calibration must be performed at a number of positions to minimize the 'drift' associated with non-linearities in mechanical devices vis-a-vis linear models for the platform and focal plane. In the testbed 1-2cm errors in absolute position and 2-3 degrees of error in absolute orientation knowledge have been observed when relying on knowledge of platform (arm) kinematics and a camera model derived in a single position camera position. In a remote servicing application, although such calibration is a ground-based function, the requirement for relocation of the telerobot at the worksite and the inevitable uncertainities in a'priori models make it impossible to expect millimeter accuracies from functions requiring

calibration. The +/-3mm accuracy in knowledge of locations between two objects reported from the relative update function has been observed in a number of camera positions in the worksite using a camera model derived in a single camera position.

Substitute human teleoperation for machine-based global path planning

Automatic planning of arm motions through the worksite requires knowledge of the location of objects as obstacles to avoid unwanted collisions with the arms. In general this is a large volume with potentially a number of objects which must be modeled, located with precision and tracked for motions during manipulations. The path planning problem is also complicated by the presence of redundant degrees of freedom for the manipulator or through a manipulator positioning system (e.g., FTS [2]). Automatic planning for such systems requires some knowledge of the task to be performed or other suboptimal constraint (e.g., position and lock of an elbow joint) to assist in selection of one solution from the many possible.

By relying on the operator to position the arms through a teleoperation function, a telerobot system does not require models of the entire worksite and relies on the operator to resolve any positional ambiguity due to the presence of redundant degrees of freedom.

Retain machine-based local path planning

The problems associated with global path planning are not as severe for local path planning: that planning of arm motions through a volume from 10's of centimeters distance until contact with the worksite. In such cases knowledge of location for a few objects is required. Such knowledge can be accurately acquired through use of operator designate and relative update. The planning of arm motions is somewhat simplified given that the resolution of ambiguity with redundant degrees of freedoms can be assumed performed through the operator teleoperation function. Lastly, the operator's visual feedback can be most restricted in near contact conditions with the worksite, making further direct teleoperation difficult if not impossible in operations with data latency on the order of a few seconds.

Although these conclusions about the use of telerobotics in remote servicing can help guide early applications of the technology, this is far from the last word on the subject. In applications where millimeter-level precision is required, a machine vision function will be better at deriving position than a human operator. In dynamically changing conditions (e.g., multiple arms/telerobots or astronauts in EVA with telerobots working at the same site) automatic tracking of locations can prevent unwanted if not unsafe collisions. With clutter in the worksite, it may not be obvious what constitutes a collision-free path through the worksite. An automatic spatial planner, used perhaps in an advisory capacity for subsequent teleoperation, can assist the operator to plan motions through the worksite. Lastly, many of the techniques discussed above rely upon an arm which can be accurately controlled, through autonomous techniques or teleoperation, to negotiate the final few millimeters and self correct through sensory feedback from encoders, force/torque sensors, etc. to allow for grasps, insertions or other manipulations. Although achievable with arms built for the commercial sector, this will, in itself, be a technology challenge in the development of a flight qualified arm which will enable the use of telerobotics for remote servicing.

CONCLUSION

This paper describes the capability of the NASA/OAST telerobot testbed in the performance of certain generic tasks, suggestive of the space assembly, maintenance and repair operations of remote servicing. Through performance in several modes: direct teleoperation with/without force reflection, shared control, traded control between teleoperation and the autonomous system, and robotic operation, the benefits of the individual technology contributions to the operation can be quantized and recommendations for use in telerobotic systems established. The development of the testbed leading to the first such quantization, using an ORU removal/replacement task, is reported. Several recommendations are offered for near-term telerobot system developments, based on the experience of this testbed project.

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the Office of Aeronautics and Space Technology, National Aeronautics and Space Administration.

REFERENCES

1. J.R. Matijevic, W.F. Zimmerman, and S. Dolinsky, "The NASA/OAST Telerobot Testbed Architecture", Proceedings NASA Conference on Space Telerobotics, Pasadena, California, January, 1989.
2. J. Andary, et.al., "The Flight Telerobotic Servicer", Proceedings NASA Conference on Space Telerobotics, Pasadena, California, January, 1989.
3. B. Hannaford, et.al., "Performance Evaluation of a Six-Axis Generalized Force Reflecting Teleoperator", JPL Publication 89-18, June 15, 1989.
4. Cepollina, F., et. al., "Extravehicular Activity Annex: Solar Max Repair Mission", NASA JSC Doc. No. 14082, Annex 11, Rev A, March, 1984.
5. "Designing an Observatory for Maintenance in Orbit: The Hubble Space Telescope Experience", Hubble Space Telescope Project Office, MSFC, Huntsville, Alabama, April, 1987.
6. "Satellite Service Handbook: Interface Guidelines", LMSC/D931647, Lockheed, December, 1983.
7. M. Montemerlo and A. DeYoung, "The Space Perspectives: Man-Machine Redundancy in Remote Manipulator Systems", Proceedings of the NATO Advanced Research Workshop on Robotics with Redundancy: Design, Sensing and Control, Salo, Lago di Garda, Italy, 1988.

THE TELEROBOT TESTBED: AN ARCHITECTURE FOR REMOTE SERVICING

SYSTEM FUNCTIONAL BLOCK DIAGRAM

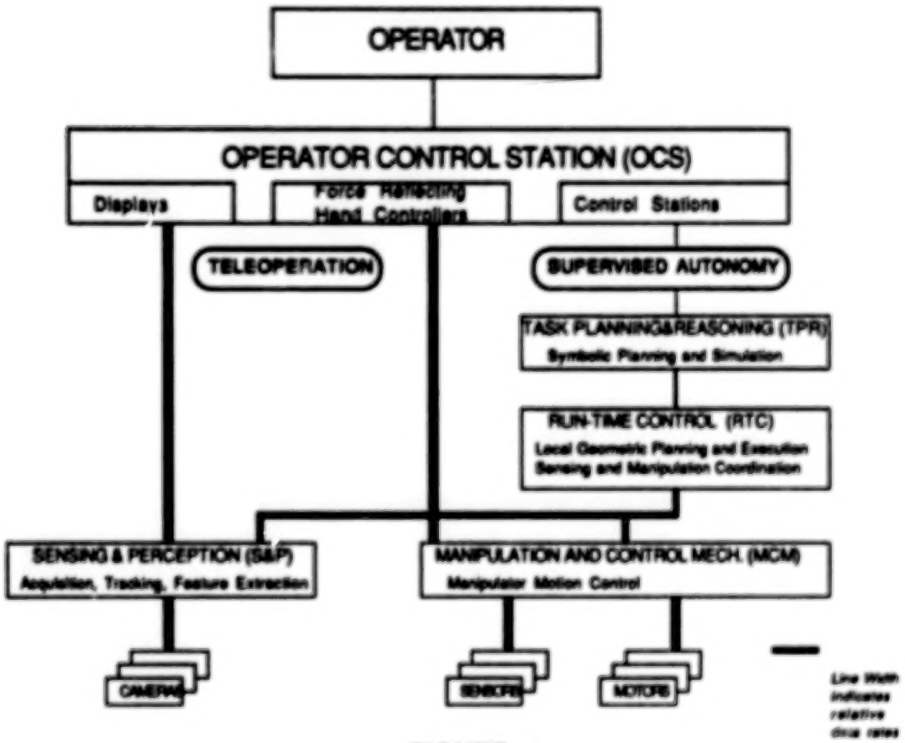


FIGURE 1

THE TELEROBOT TESTBED: AN ARCHITECTURE FOR REMOTE SERVICING

System Computing Architecture

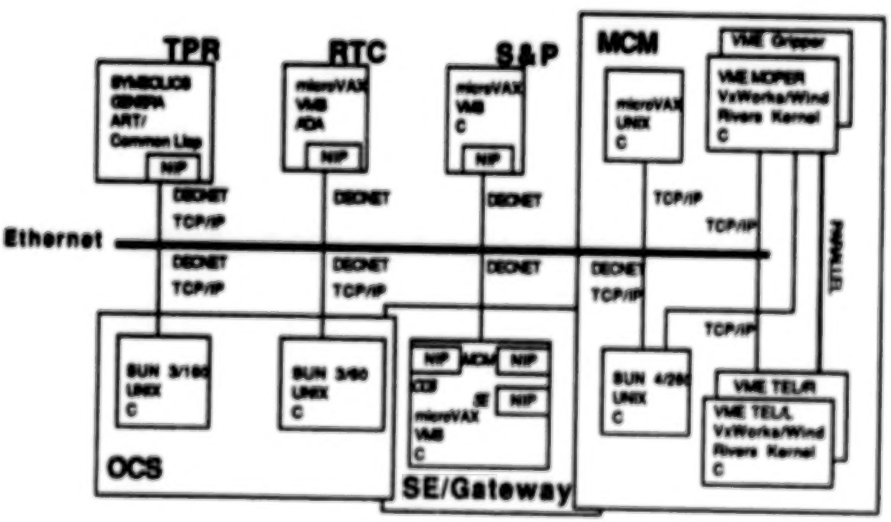


FIGURE 2

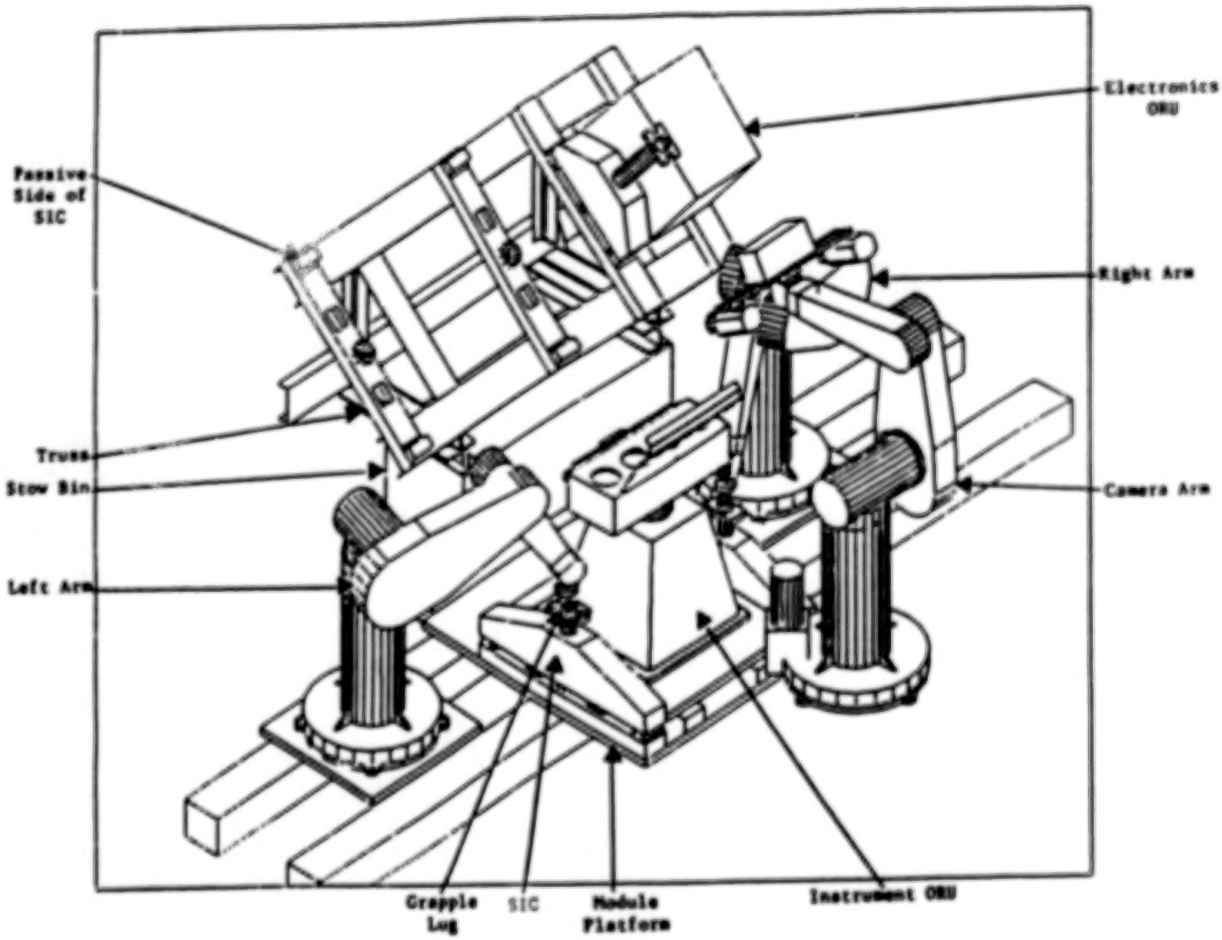

Electronics ORU
Passive Side of SIC
Right Arm
Truss
Stow Bin
Camera Arm
Left Arm
Grapple Lug
SIC
Module Platform
Instrument ORU

FIGURE 3

PERFORMANCE-BASED WORKLOAD ASSESSMENT: ALLOCATION STRATEGY AND ADDED TASK SENSITIVITY

Michael A. Vidulich
Harry G. Armstrong Aerospace Medical Research Lab
AAMRL/HEC
Wright-Patterson AFB, Ohio 45433-6573

ABSTRACT

The paper reviews the preliminary results of a research program investigating the use of added tasks to evaluate mental workload. The focus of the first studies was a reappraisal of the traditional secondary task logic that encouraged the use of low-priority instructions for the added task. It was believed that such low-priority tasks would encourage subjects to split their available resources among the two tasks. The primary task would be assigned all the resources it needed, and any remaining reserve capacity would be assigned to the secondary task. If the model were correct, this approach was expected to combine sensitivity to primary task difficulty with unintrusiveness to primary task performance. The first studies of the current project demonstrated that a high-priority added task, although intrusive, could be more sensitive than the traditional low-priority secondary task. These results suggested that a more appropriate model of the attentional effects associated with added task performance might be based on capacity switching, rather than the traditional optimal allocation model.

INTRODUCTION

Overview of Paper

The goal of the research described in the present paper was to produce guidelines for the behavioral assessment of workload. Following a few definitions, the paper will begin with a review of the conceptual underpinnings of the secondary task technique. The development of the traditional view of the secondary task technique will be reviewed, along with refinements that have recently been proposed by some authors. This will lead to a proposed alternative to the secondary task technique. The results of two experiments comparing the traditional technique to the alternative will then be presented.

Terminology. It is important to be precise about terminology. Typically, in performing a secondary task evaluation, there are two types of tasks. The first is a primary task. It is called the primary task because the subjects are told to maintain its performance at a single-task level. Typically, the primary task is the task that the researcher is interested in measuring. In an aircraft environment, a typical primary task might be maintaining flight control in a certain scenario with different possible display configurations. The secondary task is a task added by the researcher to perform the measurement. It is called secondary because the subjects are typically told to perform it as well as they can, without letting it interfere with the performance of the primary task. In other words, it is secondary in priority, relative to the primary task. The terms "primary" and "secondary" thus refer directly to the priorities the subject is instructed to assign the tasks.

Task Types. The most common pairing of task types in secondary task research (see Ogden, Levine, and Eisner (1979) for a review) has been a continuous primary task (for example, a tracking task in the lab or flight control in an aircraft simulator), paired with a discrete secondary task (for example, a Sternberg task in the lab or a communications task in a simulator). This is not the only possible combination of task types, and there is no logical reason that the assignments could not be reversed or that two tasks of the same type could not be combined. However, inasmuch as the continuous primary task and discrete secondary task has been the most common combination, it was selected for examination in the present research.

Early Single-Channel Explanations of the Secondary Task Technique

Knowles (1963) provided the first major review of secondary task research. Reflecting the dominant trend in attention theory of the time, Knowles' model of the secondary task paradigm was based on a single-channel, multiplexed information processor. In Figure 1, S1 represents the primary task's stimuli and D1 represents the primary task's decision processor. Due to instructions, or some comparable manipulation, the information processing channel would be switched to S1 and D1 whenever necessary to maintain the required level of performance. Whenever the channel was available, it would be switched to the secondary task (S2 and D2). Presumably, the more demanding the primary task, the less frequent and the shorter in duration the switches to the secondary task would be. Thus, it was expected that as primary task difficulty increased, its performance would be maintained (because of the priority instructions), but secondary task performance would degrade. In other words, the secondary task was expected to be both sensitive to primary task difficulty and unintrusive to primary task performance. Knowles acknowledged that such unintrusiveness was not common, but apparently did not think it was a major problem, so long as it was kept at a minimal level.

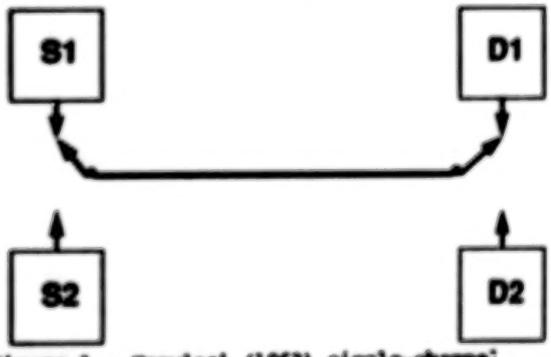


Figure 1 - Knowles' (1963) single-channel multiplexer model of the secondary task paradigm.

In 1971, Rolfe performed a review of a greatly-expanded data-base of secondary task research. As did Knowles, Rolfe explicitly argued for a single-channel model of secondary task operation. However, Rolfe also used the term "capacity," and discussed it as a divisible commodity. He described the secondary task technique as an attempt to measure the reserve capacity that was available whilst performing a primary task. Nevertheless, given Rolfe's (1971) strong endorsement of the single-channel "which must be allowed a finite time to process one stimulus-response before a second can be accepted" (p. 135), it is not clear that reserve capacity was thought to be anything more than spare time. This was consistent with Knowles' (1963) multiplexer model. Rolfe also pointed out that secondary task intrusiveness was a pervasive problem, and cautioned that primary task performance should be monitored to ensure that parity of primary task performance was maintained.

The Traditional Resource Interpretation of the Secondary Task

A major event in the interpretation of the secondary task paradigm was the publication of Kahneman's 1973 book on attention theory. The multiplexed single-channel was supplemented with the possibility of simultaneously sharing the available capacity (or "resources") among different tasks. In a number of different guises, capacity theory (or as it is also known, "resource theory") has become the central concept in discussions of the secondary task technique.

The traditional explanation of the secondary task (e.g., Williges and Wierwille, 1979) is based on the following logic: The human possesses a store of information processing resources (represented by the circle in Figure 2) that can be divided among tasks. The performance of the primary task demands some certain level of allocation from the store (represented by the shaded area). The traditional priority instructions are intended to ensure that the primary task always gets the amount of resources that it needs. Hence the name "primary." Whatever reserve capacity is left-over (indicated by the unshaded area) is allocated to the performance of the secondary task. The priority instructions are intended to ensure that the secondary task gets allocated no more than the reserve capacity of resources. Assuming the priority instructions are effective, the quality of the secondary task performance should be proportional to the size of the reserve capacity; the more reserve capacity the better secondary task performance should be. Thus, secondary task performance is expected to be sensitive to primary task demand. Also, the performance of the secondary task, if it uses only the reserve capacity, should be unintrusive to primary task performance. However, virtually every review of secondary task research since Knowles (1963) has bemoaned the fact that secondary task intrusiveness has been pervasive (e.g., Ogden et al., 1979; Rolfe, 1971; Williges and Wierwille, 1979).

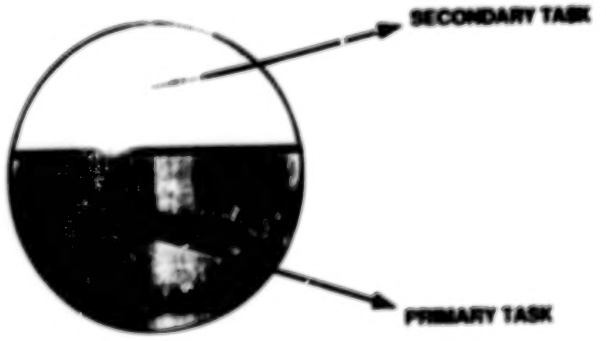


Figure 2 - The traditional capacity model of the secondary task.

Attempts to Counter Intrusiveness

Recently, there has been increasing concern about the consequences of secondary task intrusiveness. For example, Wickens (1984) cautioned that a simple comparison of the secondary task decrements that resulted from pairing with different primary tasks would be inadequate and potentially misleading. Were the subjects to differentially favor one version of the primary task less than the other primary task(s), they might inappropriately sacrifice primary task performance to maintain or improve secondary task performance. To counter this, Wickens (1984) argued that the joint performance on both the primary and secondary tasks must be compared within a Performance-Operating Characteristic (POC) space. A POC plot would have secondary task performance along one axis, and primary task performance along the other axis. Poor performance on either task would be near the origin of the figure. Thus, as joint performance of a task pair improves, the point representing that performance would move further from the origin. Also, the as the relative priorities between the time-shared tasks changes, the point will move closer to the axis representing the favored task. Therefore, plotting within the POC space provides an opportunity to compare the overall difficulty of the different primary tasks, even in situations in which parity of intrusiveness was not maintained.

Going even further with this logic, Gopher and Donchin (1986, p. 41-26) suggested that,

> Maximization of interference appears to be more consistent with the original secondary task rationale, in which the second task is added to saturate the capacity of the system, create an overload, and enable one to scale the demands of the primary task. It is, therefore, somewhat surprising that a lack of obtrusiveness of the introduction of a secondary task to the performance of a primary task has been identified as a highly desired property of a good secondary task.... How can this aspiration coexist with the main thrust of a technique that advocates the study of interference patterns as its main tool?

Gopher and Donchin (1986) pointed out that the traditional view of the secondary task made the strong, but questionable, assumption that subjects had full voluntary control over their resource allocation. As a result of problems with this, and other assumptions underlying the secondary task technique, Gopher and Donchin proposed replacing the traditional secondary task approach with full POC methodology. That is, instead of specifying a single set of priorities, subjects would be instructed to perform trials with a variety of relative priorities between the tasks. The use of multiple priorities would generate a sufficient number of data points for the generation of complete POCs outlining the performance tradeoff function between the two tasks. In contrast, Wickens' (1984) discussion of the secondary task technique was limited to the usual instructions, but evaluated the results within a POC space.

However, there are potential problems with the use of POC methodology as a workload assessment tool. For example, the use of multiple instruction sets requires an expansion of the experimental design. The minimum number would be two levels of relative priority for each task pair, which would double the size of any experiment relative to a single-point secondary task evaluation. This would, of course, be very expensive and time-consuming. Second, there is the issue of complexity. Complexity can refer to both the more complicated experimental design, which is likely to be unattractive to system evaluators who do not come from a background of attention research, and also to the more complicated data demands of full POC methodology. Unlike the traditional secondary task technique, plotting within a POC space requires a measure of primary task performance. Such a measure is certainly desirable, whenever it is available. But in complex operational tasks a measure of primary task performance might be unobtainable, or perhaps even difficult to define. Also, with the increasing trend towards automated systems, it is likely that there may be a need to assess workload in environments in which very few responses are ever made to the "primary" task. A third problem is operator acceptance. In operational settings, researchers often try to "hide" the secondary task as one of the tasks within a complex, but realistic, set of tasks. If subjects are told that one aspect of a task that they normally perform is changing priorities over trials, it might compromise the realism and reduce the operator acceptance.

Of course, if Gopher and Donchin are correct about the deficiencies of the traditional secondary task technique, then all of the problems associated with POC methodology will have to be accepted and applications constrained to environments allowing full POC methodology. But, the problems with using POC methodology as the standard technique are dire enough to provide serious encouragement for investigating other alternatives.

An Alternative Model of Secondary Task Operation

The present research was designed to study a possible alternative to POC methodology. But, before getting into the alternate approach or the research that was performed, it is advisable to alter the terminology somewhat. Instead of the usual primary/secondary distinction, the designation of the tasks will be separated from any explicit indication of their relative priorities. One task will be referred to as the "measured" task (i.e., the task whose demand must be measured). The other task will be referred to as the "added" task (i.e., the task that was added to provide a measure of the

measured task's workload). The reason for this change should become obvious in a moment.

The traditional secondary task paradigm assumed that subjects could voluntarily control their resource allocation finely enough that the secondary task would have access to only the reserve capacity. However, the pervasiveness of intrusions in the previous secondary task research indicates that the subjects' control is not as flexible as expected. There have been numerous demonstrations that a sudden discrete stimulus can pre-empt the processing of a time-shared continuous task (e.g., Klapp, Kelly, and Hetick, 1987; McLeod, 1977). So perhaps, instead of using only the reserve capacity for an added task, the subjects momentarily withdraw their resources from the measured task and reallocate them to the added task. This is illustrated in Figure 3 by the arrow moving from the measured task to the added task, implying that the discrete added task momentarily gains access to all resources.

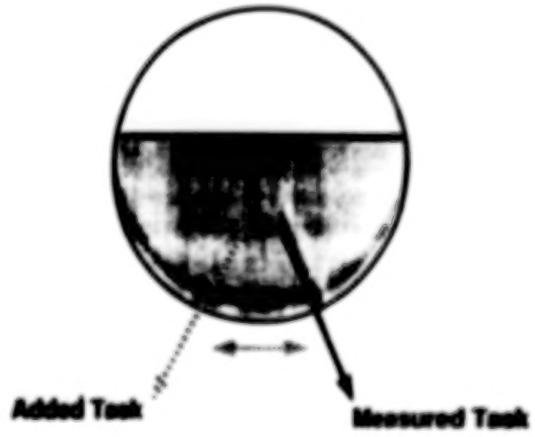


Figure 3 - An alternative model of added task methodology based on capacity switching.

This may appear to be very reminiscent of Knowles' (1963) single-channel interpretation, but there is a difference. In Knowles' model the switch was a structural feature, located outside the processors. The time taken for the switch to move from one task to another would be expected to be independent of the individual tasks' parameters. But, within what could be called a "capacity switch" model, it is reasonable to imagine that the amount of time required to reallocate resources to a new task might be influenced by the level of allocation to the current task. So, the more difficult the measured task, the longer it would take for the added task's pre-emption to be accomplished. Within this viewpoint, measured task intrusiveness is inescapable, but does not mitigate against sensitivity. The best way to measure the time required by the switch would be to instruct subjects to switch as quickly as possible to the added task, in the expectation that the switch would be slower as measured task difficulty increased. In other words, rather than the traditional low-priority secondary task, there would be a high-priority added task.

EXPERIMENT ONE

These two sets of priority instructions were tested in a laboratory experiment. The tasks were selected from the Air Force Criterion Task Set (Shingledecker, 1984). The measured task was a continuous unstable tracking task. An unstable tracking task is akin to trying to balance a vertical stick on the end of your fingertips. As the tracking task's lambda parameter is increased, the task becomes more unstable. The effectiveness of lambda as a workload manipulation, had been demonstrated in earlier laboratory experiments that studies the lambda effects on root mean square (RMS) error and subjective workload ratings (Shingledecker, 1984).

The added task was a Sternberg memory search task. The memory set was either 2 or 4 letters. Probe letters appeared periodically during the course of the trial. The average interstimulus interval (ISI) was 3 s, but the ISI varied randomly from 2.5s to 3.5s to produce some temporal uncertainty.

There were 20 subjects in the experiment. The subjects were recruited from local colleges and paid for their participation.

Each subject performed the dual-tasks under two sets of instructions. In one session the Sternberg task was the high-priority task. The subjects were instructed to track as well as they could when no Sternberg probe was present, but to consider the Sternberg the most important task whenever one of its probes appeared. In another session, the subjects were told the Sternberg was the low-priority task. They were asked to perform the Sternberg as well as they could, without letting it interfere with tracking performance. This corresponds to the usual secondary task instructions.

The question the experiment was designed to address was; "Which instructions would produce the Sternberg performance that was most sensitive to tracking difficulty?" That is, under which set of instructions would Sternberg reaction time be most affected by tracking lambda?

Figure 4 displays the reaction time (RT) results as a function of lambda, for the two sets of instructions: The high-priority Sternberg results are on the left, and the low-priority Sternberg results are on the right. In neither case, were the Sternberg reaction time data sensitive to the tracking task's lambda. This result was not anticipated. However, an interpretation of this result is possible within multiple resource theory.

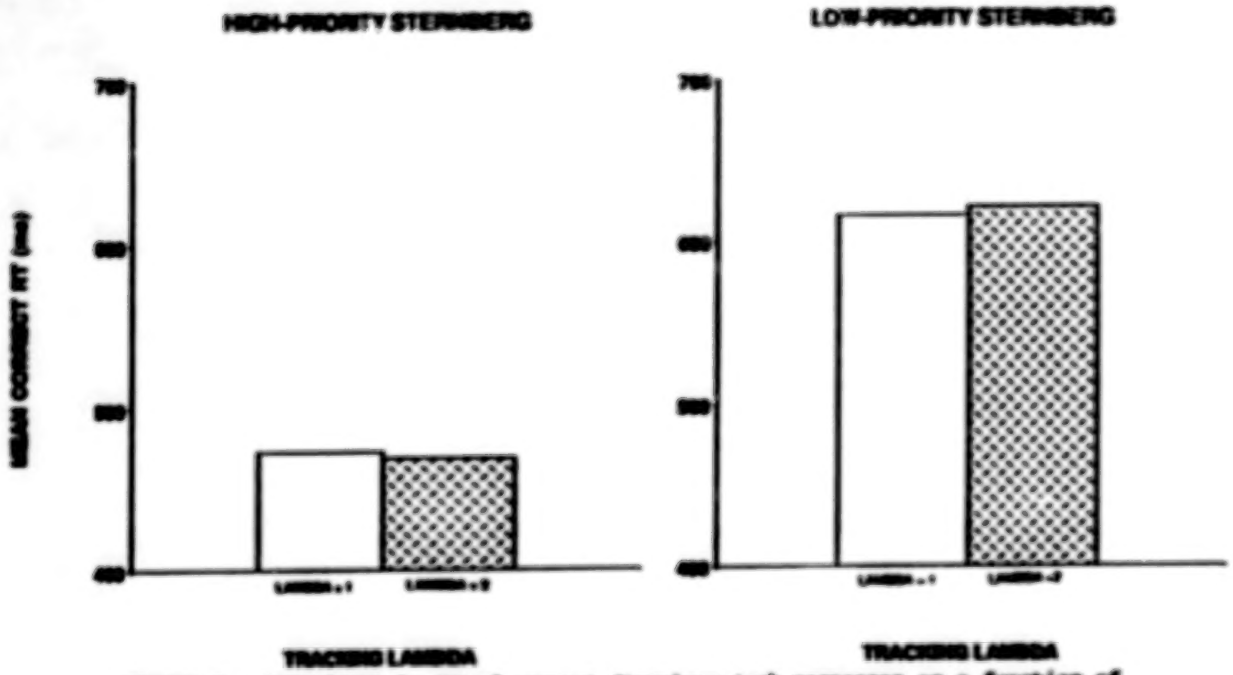


Figure 4 - Experiment 1, RT of correct Sternberg task responses as a function of priority and tracking lambda.

Wickens' (1980) multiple resource model distinguishes between early, perceptual/central processing resources and late, response execution processing resources. The tracking task used an easily-seen, highly-compatible display with no real need for predicting future cursor actions (as would be required in higher-order tracking). So, the perceptual/central demands of this tracking task were probably very low, regardless of the lambda level. But, the tracking did require the generation of frequent gradual responses. Increasing lambda to 2 would increase the frequency of these responses, adding even more to the high response execution demands, but possibly without any significant increase in the perceptual/central demands.

On the other hand, the response execution demands of the Sternberg responses were probably trivial, because the subjects only needed to press a button. But, the scanning of the memory set would be expected to place relatively heavier demands on perceptual/central resources, and these demands would be expected to increase as the memory set was expanded from 2 to 4 items.

So, it is plausible that the two tasks simply failed to overtax any single source of resources. To test this hypothesis, the Sternberg task was redesigned to compete more directly with the tracking.

EXPERIMENT TWO

In Experiment 2, the measured task was still the unstable tracking task with two levels of lambda. However, the Sternberg task was changed. Instead of manipulating the Sternberg task's memory set size (it was set to 3), the response device was manipulated. On half of the trials, the subjects used the same button box that was used in Experiment 1, but on the other half of the trials the subjects had to deflect a joystick in the appropriate direction and press a button on top. Not only did this joystick response require a greater muscular involvement than the button-press; but also, because the joystick was on an easily-tipped stand, it required a very precise movement, as well. Changing the Sternberg response device from button to joystick was intended to directly compete with the tracking task's demand for response execution resources.

The same priority manipulation used in Experiment 1 was used in Experiment 2. The experimental questions were now twofold: One, would the control manipulation increase inter-task interference enough to generate sensitivity? And, Two, if it did, which instructions would produce the most sensitive Sternberg task performance?

There were 26 subjects in the experiment. The subjects were recruited from local colleges and paid for their participation.

In Figure 5, it can be seen that the manipulation was effective. On the left side of Figure 5 there is a small, but statistically significant, effect of lambda on the Sternberg reaction time. When subjects were given the high-priority Sternberg instructions, RT to the Sternberg task was 11 ms faster when time-shared with the easy lambda tracking task than with the difficult lambda tracking task ($F(1,25) = 4.69$, $p < 0.04$). On the right side of Figure 5, it is clear that the low-priority Sternberg is still insensitive to tracking demands.

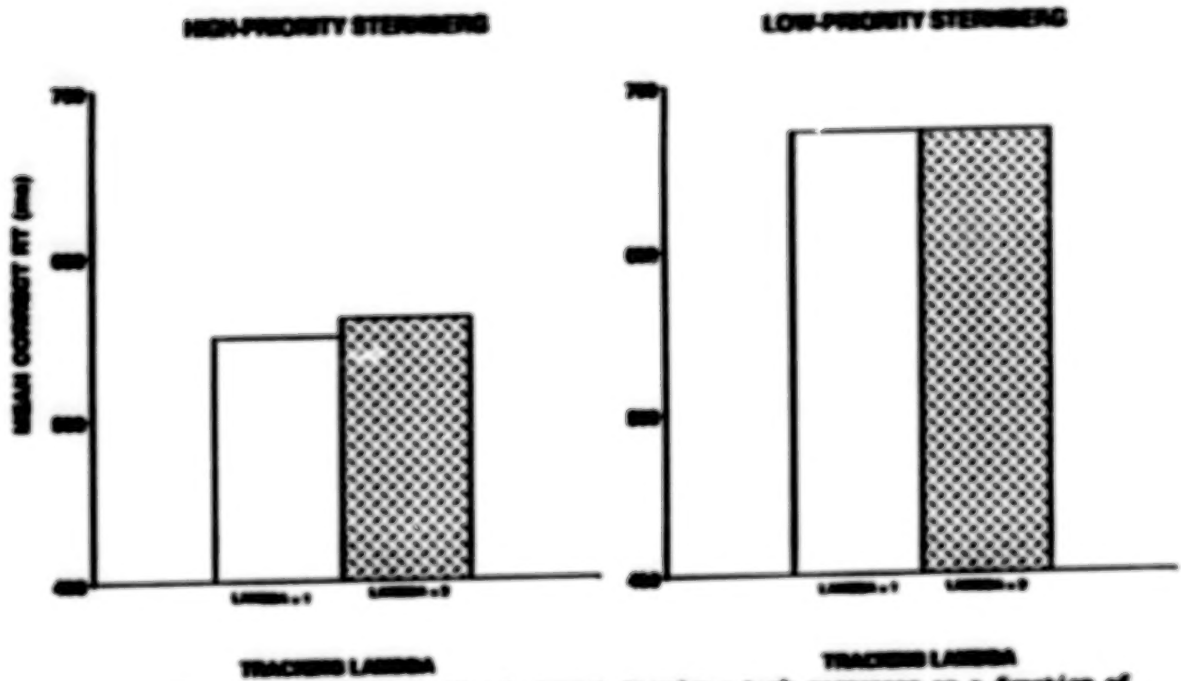


Figure 5 - Experiment 2, RT of correct Sternberg task responses as a function of priority and tracking lambda.

But, what about intrusiveness? To compare the intrusiveness caused in tracking performance by the occurrence of the Sternberg probe under the different priority instructions, the tracking RMS error was analyzed by intervals around the Sternberg probe occurrence time (see Figure 6). In the center of each figure, is the one-second interval during which the Sternberg probe was presented. The two one-second intervals directly before each Sternberg probe presentation, and the two intervals directly after were also recorded for analysis. In the analysis, a significant Priority x Sternberg Control x Interval interaction ($F(4,188) = 5.78$, $p < 0.001$) illustrated the effects of the variables.

In Figure 6, it can be seen there was little evidence of intrusiveness in the low-priority Sternberg condition (on the right side of the figure). The RMS error was virtually flat across intervals in the button trials (the solid line), and showed only a slight effect in the joystick trials (the dotted line). The left side of the figure, displaying the high-priority Sternberg results, looks quite different. The overall rise in the RMS error was a by-product of the priority instructions emphasizing the Sternberg task, at the expense of the tracking. But, more important is the shape of the curves. The "U" shape may seem a bit peculiar at first, but the heightening of each "U's" left side was probably an artifact of too short an ISI. The average ISI was only 3 s, and it appears that the subjects were just recovering from one Sternberg stimulus when the next stimulus was presented. In any case, it can be seen that immediately following a Sternberg stimulus presentation, there was a rise in RMS error, and this rise appears to be steeper when the Sternberg response control was the joystick. So, the Sternberg control manipulation was successful in increasing interference with the tracking task, and this increased interference was correlated with greater sensitivity in the high-priority Sternberg condition.

In Experiment 1, this three way interaction was not statistically significant. The Experiment 1 data, when plotted in the same manner as Figure 6, generated flat lines across the five intervals, regardless of priority or memory set size. In Experiment 1, the two tasks' difficulty manipulations simply did not conflict with each other.

(Note - The method and results for Experiment 2 have been also discussed in Vidulich (in press)).

CONCLUSIONS

Taking all of the results into consideration encourages the following conclusions:

(1) The high-priority added task appears to be a viable approach for workload assessment. It appears to be more sensitive than the traditional low-priority secondary task approach, and is much less demanding to implement than full PDC methodology.

(2) The sensitivity of an added task appears to be directly related to its intrusiveness. In cases where the added task did not intrude on the measured task's performance, it was also insensitive to the measured task's difficulty.

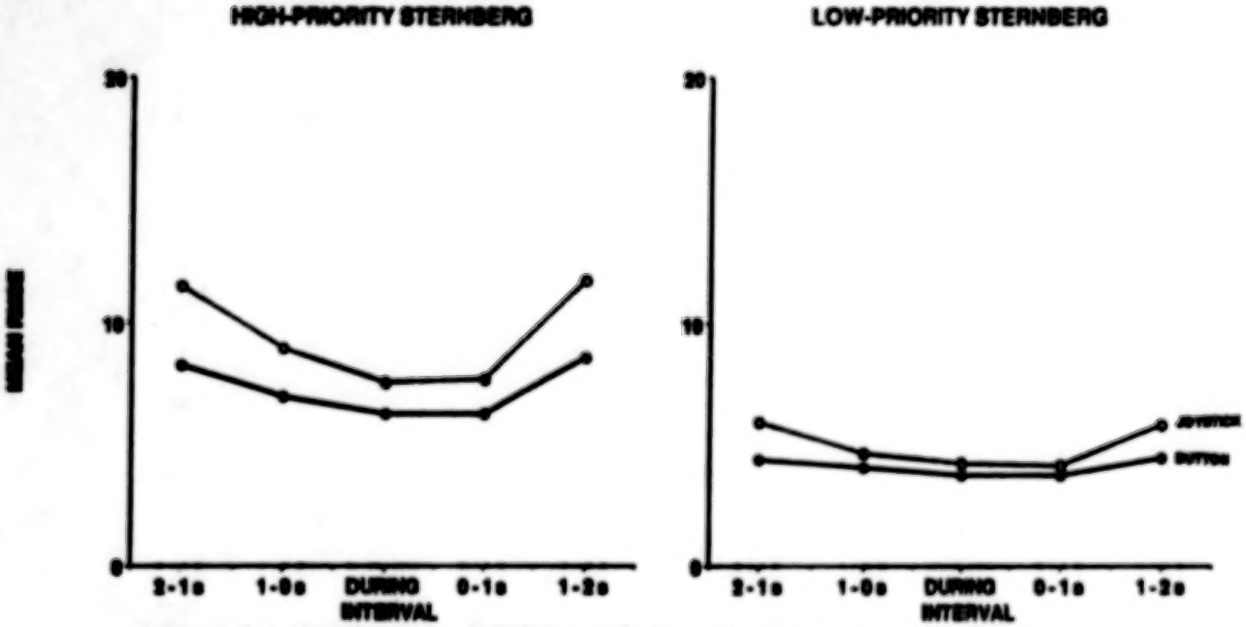


Figure 6 - Experiment 2, Sternberg task intrusiveness in tracking RMS error as a function of interval, priority, and Sternberg task control device.

And, (3) Intrusiveness seems to depend on specific forms of interference. In Experiment 1, the two tasks were too different in their processing needs, and the level of interference was insufficient to generate sensitivity, regardless of instructions.

These two studies are an insufficient basis for an unqualified endorsement of the high-priority added task alternative. But, the results are certainly encouraging enough to encourage further research. A third experiment in this series is currently being planned, in Experiment 3 an attempt will be made to modify the tracking task to make it compete more with the original Sternberg memory set size manipulation. Planned follow-ups will include study of the effects of predictability of added task time of occurrence and continuous added tasks. Either of these manipulations might reduce the unexpected abruptness of the added task occurrence and make the situation more amenable to finer voluntary control of resource allocation. If so, the traditional low-priority secondary approach may perform better.

REFERENCES

Gopher, D., and Donchin, E. (1986). Workload - An examination of the concept. In K.R. Boff, L. Kaufman, and J.P. Thomas (Eds.), Handbook of perception and human performance - Volume II, Cognitive process and performance (pp. 41-1 - 41-49). New York: Wiley.

Kahneman, D. (1973). Attention and effort. Englewood Cliffs, NJ: Prentice-Hall.

Klapp, S.T., Kelly, P.A., and Netick, A. (1987). Hesitations in continuous tracking induced by a concurrent discrete task. Human Factors, 29, 327-337.

Knowles, W.B. (1963). Operator loading tasks. Human Factors, 5, 155-161.

McLeod, P. (1977). A dual-task response modality effect: Support for multiprocessor models of attention. Quarterly Journal of Experimental Psychology, 29, 651-667.

Ogden, G.R., Levine, J.M., and Eisner, E.J. (1979). Measurement of workload by secondary tasks. Human Factors, 21, 529-548.

Rolfe, J.M. (1971). The secondary task as a measure of mental load. In W.T. Singleton, J.G. Fox, and D. Whitfield (Eds.), Measurement of man at work (pp. 135-148). London: Taylor and Francis.

Shingledecker, C.A. (1984, November). A task battery for applied human performance assessment research (Tech. Rep. No. AFAMRL-TR-84-071). Wright-Patterson AFB, OH: Air Force Aerospace Medical Research Lab.

Vidulich, M.A. (in press). Objective measures of workload: Should a secondary task be secondary? In Proceedings of the Fifth International Symposium on Aviation Psychology.

Wickens, C.D. (1980). The structure of attentional resources. In R. Nickerson (Ed.), Attention & performance VIII (pp. 239-257). Hillsdale, NJ: Erlbaum.

Wickens, C.D. (1984). Engineering psychology and human performance. Columbus, OH: Merrill.

Williges, R.C., and Wierwille, W.W. (1979). Behavioral measures of aircrew mental workload. Human Factors, 21, 549-574.

THE WORKLOAD OF SUSTAINED ATTENTION

Dr. P. Hancock
University of Southern California

J. Warm, W. Dember, J. Gluckman, T. Galinsky, J. Thremann, and A. Becker
University of Cincinnati

(Paper not provided by publication date.)

REAL-TIME MEASUREMENT OF MENTAL WORKLOAD: A FEASIBILITY STUDY

Arthur Kramer, Darryl Humphrey, Erik Sirevaag & Axel Mecklinger

Department of Psychology
University of Illinois

The primary goal of our study was to explore the utility of event-related brain potentials (ERP) as real-time measures of workload. To this end, subjects performed two different tasks both separately and together. One task required that subjects monitor a bank of constantly changing gauges and detect critical deviations. Difficulty was varied by changing the predictability of the gauges. The second task was mental arithmetic. Difficulty was varied by requiring subjects to perform operations on either two or three columns of numbers. Two conditions that could easily be distinguished on the basis of performance measures were selected for the real-time evaluation of ERPs. A bootstrapping approach was adopted in which one thousand samples of n trials (n = 1, 3, 5 ...65) were classified using several measures of P300 and Slow Wave amplitude. Classification accuracies of 85% were achieved with 25 trials. Results are discussed in terms of potential enhancements for real-time recording.

INTRODUCTION

The research presented here derives from an extensive series of investigations that have demonstrated the utility of Event-Related Brain Potentials (ERPs) in the assessment of residual capacity during the acquisition and performance of a variety of perceptual-motor and cognitive tasks (Donchin et al., 1986; Kramer, 1987). The focus of the present study was to assess the feasibility of employing ERPs as on-line measures of mental workload. If physiological data, and ERPs in particular, are to serve as real-time measures of operator mental load, the amount of data (e.g. secs, mins?) necessary to reliably discriminate among levels of workload must be determined. This question will be addressed in the present study by adopting a bootstrapping approach in which we examine the classification accuracy of ERP measures with from 1 to 65 secs of data. However, before we describe our experiment in detail we will briefly discuss the previous research that suggests that ERPs provide a sensitive and reliable measure of mental load in an off-line context.

Several recent studies have illustrated the usefulness of the ERP, and more specifically the P300 component, as an index of processing resources (Horst et al., 1984; Isreal et al., 1980; Kramer et al., 1985, 1987; Natani and Gomer, 1981; Sirevaag et al., 1988). The general paradigm employed in these studies requires subjects to perform two tasks concurrently. One task is designated as primary and the other task as secondary. Subjects are instructed to maximize their performance on the primary task and devote any additional resources to the performance of the secondary task.

ERPs are elicited by events in either one or both of the tasks. Increases in the perceptual/cognitive difficulty of the primary task result in a decrease in the amplitude of the P300s elicited by the secondary task. Conversely, P300s elicited by discrete events embedded within the primary task increase in amplitude with increases in primary task difficulty. Furthermore, changes in response related demands of a task have no influence on the P300 (Isreal et al., 1980).

The reciprocal relationship between P300s elicited by primary and secondary task stimuli is consistent with the resource tradeoffs presumed to underlie dual-task performance decrements (Kahneman, 1973; Navon and Gopher, 1979; Sanders, 1979; Wickens, 1980). That is, resource models predict that as the difficulty of one task is increased, additional resources are re-allocated to that task in order to maintain performance, thereby depleting the supply of resources that could have been used in the processing of other tasks. Thus, the P300 appears to provide a measure of resource tradeoffs that can only be inferred from more traditional performance measures. Furthermore, P300s elicited by secondary task events are selectively sensitive to the perceptual/cognitive demands imposed upon the operator. This selective sensitivity may be especially useful in decomposing the changing processing requirements of complex tasks (Kramer, 1987).

One might ask why ERPs should be used to monitor changes in resource demands given that several technically simpler approaches to the assessment of skill acquisition and mental workload have already been implemented. Although

numerous performance-based measures of mental workload exist, they suffer from several drawbacks. First, some of the measurement techniques require subjects to perform a secondary task which frequently interferes with the performance of the task of interest (Knowles, 1963; Rolfe, 1971; Wickens, 1979). This is clearly unacceptable in an operational environment in which the safety of the operator must be assured. Even in the laboratory setting it is difficult to determine which of the two tasks generated an observed performance decrement since the performance on the two tasks is easily confounded. Second, performance-based measures of mental workload provide an output measure of the operator's information processing activities (e.g. RT, accuracy). Thus, at best, performance measures provide only an indirect index of cognitive function. Third, performance measures do not always correlate highly with the actual workload of the tasks (Brown, 1978; Dornic, 1980; Ogden et al., 1979). Fourth, although subjective measures are relatively easy to collect and possess high face validity they do not reflect the moment to moment variations in workload that can be indexed by physiological measures.

The present study is part of a continuing effort to explore the utility of psychophysiological measures of mental workload. A primary aim of the project is to determine the feasibility of on-line uses of integrated psychophysiological and performance data. However, given the magnitude of the project this report will be confined to a description of a preliminary examination of signal/noise ratio parameters of ERPs. More specifically, we will derive the functions that relate amount of ERP data to discrimination accuracy between workload conditions.

METHODS

Subjects

Four dextral subjects (2 female) were paid $4.00/hour plus a dollar/day bonus for their participation in five sessions. All subjects had normal or corrected-to-normal vision.

Tasks

Two different tasks were performed both separately and together. We will describe each of the tasks in detail.

Monitoring Task. One task consisted of monitoring six gauges. The behavior of a gauge was determined by the interaction of four properties: update speed, noise level, noise frequency and transients. The cursors moved around the gauges at different speeds, a slower gauge taking longer to reach the critical region. Noise level was the amount ofrandom jitter in the cursor. Noise frequency determined how often random fluctuations were added to a gauge. The addition of transients also served to perturb a gauge.

The interaction of these properties produced cursor driving functions of varying predictability. Manipulating the driving functions allowed control over gauge monitoring difficulty. The driving functions employed in the high predictability (HP) conditions were such that within a row of three gauges the driving functions were identical in terms of speed, noise level, and noise frequency; no transient occurred for any gauge. The two rows differed in the speed of cursor movement, speed being constant within a row. For the low predictability (LP) conditions the average value for all properties was equivalent to the HP conditions, however, the individual values were varied with no established correlation between any set of gauges. The LP conditions contained three gauges with a transient. The frequency of the transient was different for each of the three gauges.

The gauges were presented on a CRT in front of the subject. Each gauge was divided into 12 regions (labelled 1 to 12). In addition, each third of the gauge was distinctly colored (green, yellow and red). The critical level was designated by the position marked by the numeral 9, which was the first region in the red zone.

The purpose of this task was to reset each gauge as quickly as possible once its cursor had entered the critical region. To reset a gauge the subjects pressed one of six keys after which the cursor returned to the starting position marked by the numeral 1. The cursors were not continuously visible. To sample a given gauge the subject pressed one of a set of six keys with their left hand. The cursor remained visible for 1000 msec. Simultaneous sampling was not possible.

Mental Arithmetic Task. The center of each gauge served as a display area for the operands and operators of the mental arithmetic trials. All of the operands and operators were presented simultaneously and remained in view until an answer was entered or for a maximum of 30 seconds. An answer window appeared to the right of the gauges. Answers were entered via the numeric keypad of the response keyboard and appeared in the window as they were typed. Completion was signaled by pressing the 'enter' key of the numeric keypad. The inter-trial interval varied from four to fifteen seconds. Difficulty was manipulated by varying the number of

column operations necessary to complete the problem. The easy version of the task required operations on two columns while the difficult version of the task required operations on three columns of numbers. Henceforth, these versions of the tasks will be referred to as A2 and A3, respectively. Operations included addition and multiplication.

Subjects participated in five sessions. The first two sessions constituted training. Single task conditions, starting with the easy conditions progressing to the difficult conditions were performed first, followed by the dual task conditions. In the final three sessions the subject performed the eight conditions in a random order determined by a Latin square design. Only the data from the last three sessions will be presented in this report. In all sessions two blocks of each condition were run consecutively, each block taking five minutes. A five minute break was imposed at the halfway point in addition to any breaks the subject requested.

Performing the gauge monitoring and mental arithmetic tasks in all possible combinations yields eight conditions: 2 task types X 2 levels of difficulty X 2 task pairings (single or dual task condition).

ERP Recording

Electroencephalographic (EEG) activity was recorded from three midline sites (Fz, Cz, Pz according to the International 10-20 system: Jasper 1958) referenced to averaged mastoids. All electrodes were Sensormedics Ag/AgCL electrodes. The scalp electrodes were affixed with Grass EC2 electrode cream. The forehead ground, mastoid and electrooculgram (EOG) electrodes were affixed with the Grass cream and electrode collars. Vertical and horizontal EOG was in order to control for eye movement artifacts. Electrode impedance was maintained below 10 kohms.

The EEG and EOG were amplified by Grass 12A5 amplifiers with a 8 sec time constant and a low-pass filter of 100 Hz. The recording epoch was 1300 msec beginning 100 msec prior to an event. The data channels were digitized every 5 msec and were filtered off-line (-3 db at 6.89 Hz., 0 db at 22.22 Hz) prior to further analysis. The psychophysiological data collection was governed by a DEC PDP 11/73 computer system. Artifact rejection was based upon the vertical eye movement standard deviation. ERPs were recorded during the three experimental sessions.

Subjects were seated in a dimly lit, sound attenuated booth. Stimuli were presented on a color monitor located 80 cm in front of the subject. Stimulus presentation and behavioral data collection were performed by an IBM AT.

Data Analysis Procedures

ERP eliciting events included critical gauge samples, non-critical gauge samples and, presentation of math trials. ERP measurements included P300 latency, P300 base-to-peak amplitude, P300 base-to-peak area and, slow wave area. Behavioral variables included accuracy and response speed in both the monitoring and arithmetic tasks.

In an effort to determine the amount of physiological data needed to discriminate among different experimental conditions we applied a bootstrapping approach to single trial ERP data. Given the amount of data collected in our study we decided to begin by examining the physiological differences between two conditions that could be discriminated on the basis of performance measures: the LP single task gauge condition and the gauge samples from the LP/A3 dual task conditions. One thousand samples of size n (n = 1,3,5,...,65) were randomly selected from single trial data in each of these conditions. By comparing the single trial samples with the grand average waveforms for that condition the single trial may be classified as a hit (belonging to the criterion condition), a miss (not belonging to the criterion condition) or unclassifiable. Tabulating the classification results in a 2 X 2 contingency table enabled us to assess the efficiency of a number of ERP measures.

RESULTS & DISCUSSION

The results will be organized in the following manner. First, we will describe the effects of single and dual task manipulations on subjects' performance and ERPs. These analyses will enable us to establish the relative differences in performance and workload among the single and dual task conditions. Second, we will select two experimental conditions that can be distinguished on the basis of average performance and ERP measures. A bootstrapping approach will then be applied to the single trial ERP data in these conditions. The classification accuracy value derived from each sample of one thousand measures will then be plotted as a function of the number of trials in each of the thousand samples. This procedure enables us to determine how changes in the signal/noise ratio of the ERP as a function of averaging (e.g. averaging from 1 to 65 trials for each of the thousand samples) translates into gains in the accuracy of discrimination between workload conditions.

The bootstrapping approach will be applied to several different ERP measures including: base to peak measures of P300 amplitude (P3bp), measures of P300 area (P3area), cross-correlation measures of P300 amplitude (P3cross), and area measures of a late slow wave component (SWarea). P3bp was defined as the largest positivity in the waveform between 300 and 800 msec post-stimulus relative to a pre-stimulus baseline. The "stimulus" was the presentation of the cursor with the gauges. P3area was defined as the area in a 100 msec window centered around the peak. P3cross measures were calculated by moving a 300 msec wide cosine wave across the period from 300 to 800 msec post-stimulus. The slope of the regression function at the point at which the correlation between the cosine "template" and the ERP waveform was maximized was defined as P3cross. SWarea was defined as the area between 750 and 1000 msec post-stimulus.

Effects of Experimental Manipulations

Figure 1 presents a measure of the accuracy with which subjects reset the gauges in each of the monitoring conditions. A "miss" was scored when subjects failed to reset a gauge within 10 sec following the point at which it reached a critical value. As can be seen from the figure, accuracy decreased from single to dual task conditions and again with an increase in the difficulty of the dual task. Accuracy also appeared to differ as a function of the predictability of the gauges (HP vs. LP). These differences were confirmed by a repeated measures 2-way ANOVA, with gauge (2 gauge conditions, HP and LP) and task (3 arithmetic conditions, none, A2 and A3) as factors. Significant main effects were obtained for both the gauge ($F(1,3)=13.2$, $p<.01$) and task ($F(2,6)=21.2$, $p<.01$) factors. A marginally significant interaction between gauge and task factors was also obtained ($F(2,6)=2.9$, $p<.08$) suggesting a decrease in accuracy at the most difficult level of each of the factors.

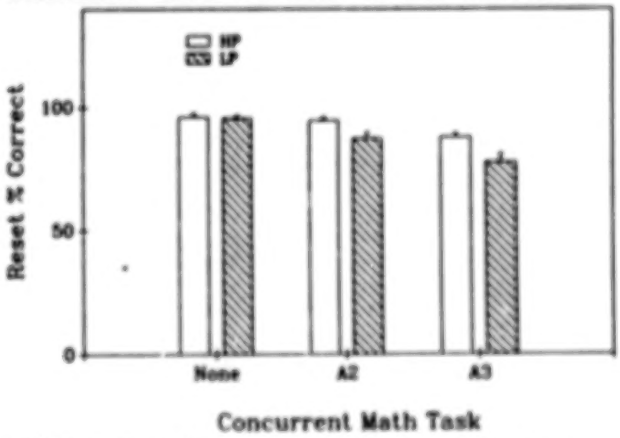


Figure 1. Accuracy in the monitoring task.

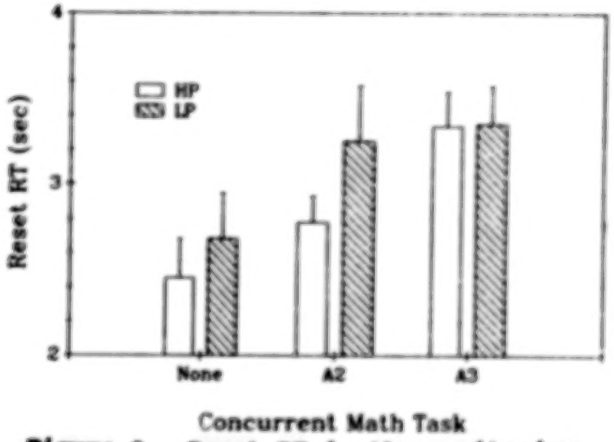


Figure 2. Reset RT in the monitoring task.

Figure 2 presents gauge reset RTs for each of the monitoring conditions. A repeated measures ANOVA performed on this data set revealed a significant main effect for the task factor ($F(2,6)=5.4$, $p<.01$). RT increased from the single to the dual task conditions and again from the A2 to the A3 versions of the arithmetic task. The main effect for the gauge factor did not attain statistical significance.

Accuracy and RT measures are presented for the arithmetic task in figures 3 and 4, respectively. Accuracy in the arithmetic task was higher when operations were performed on two columns than when a three column problem was performed ($F(1,3)=22.8$, $p<.01$). RT was also faster in the A2 than in the A3 version of the arithmetic task ($F(1,3)=26.4$, $p<.01$). Finally, RT in the arithmetic task increased with the transition from the single to dual task conditions and again when the difficulty of the monitoring task was increased.

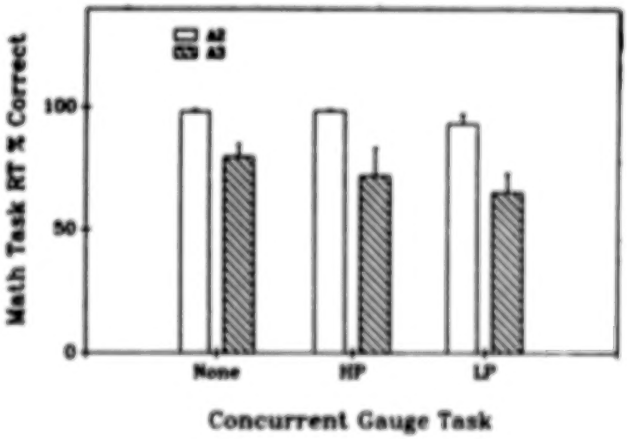


Figure 3. Accuracy in the arithmetic task.

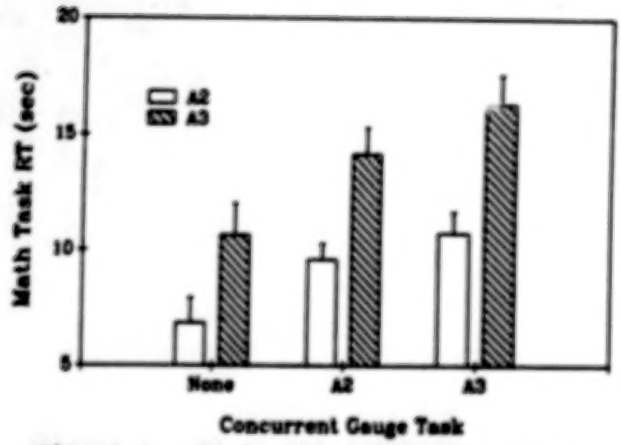


Figure 4. RT in the arithmetic task.

Real-Time Analysis of Mental Workload

Given the substantial amount of analysis time required to perform the "bootstrapping" operation we decided to select two experimental conditions to analyse further. In order to perform the bootstrapping operation it was necessary for the experimental conditions to meet three criteria. First, there should be a substantial number of trials available in the selected conditions. This was necessary since repeated samples of 1000 trials would be selected during the bootstrapping operation. Second, the conditions should be discriminable on the basis of performance measures. Thus, we wanted to begin our analysis of the real-time potential of ERPs by selecting two clearly discriminable conditions. Later analyses will examine conditions that are less discriminable. Third, the conditions should be discriminable on the basis of average ERP measures. Based on these criteria we selected two conditions from the monitoring task: the single task LP condition and the dual task LP/A3 condition.

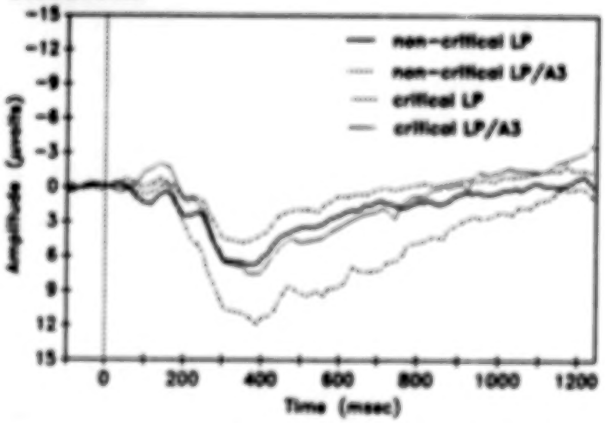


Figure 5. Grand average ERPs recorded at Pz for four of the monitoring conditions.

Figure 5 presents the grand average ERPs across the four subjects for the LP and LP/A3 conditions. It is important to note that we have further subdivided the conditions into waveforms that were elicited during times at which the gauges were in the acceptable range and other times in which the gauges had gone critical. Since the gauge critical samples were most closely associated with the performance measures we decided to employ ERPs to discriminate between the LP and LP/A3 conditions during the gauge critical periods. Approximately 200 trials were available in each of these conditions for each of the subjects. The bootstrapping operation was performed separately on the data from two of the original four subjects.

As described above, the bootstrapping operation involved the repeated selection of single trial ERPs from each of the conditions. Each "sample" was comprised of 1000 ERP measures, 500 selected from the LP condition and 500 selected from the LP/A3 condition. Each of the ERP measures was composed of an average of from 1 to 65 single trial ERP waveforms. Classification accuracy was determined by computing the relative "distance" of each ERP measure from the subject's grand average ERP measures in the LP and LP/A3 conditions. For example, if a subject possessed a grand average P300 amplitude of 50 microvolts in the LP/A3 condition and 10 microvolts in the LP condition then a single trial measure of 46 microvolts would be classified as LP/A3. This classification procedure was performed for each of the 1000 ERP measures in a sample and for each of the different pattern recognition techniques (i.e. P3bp, P3area, P3cross, SWcross).

Figures 6 and 7 present the classification functions for subjects 2 and 3, respectively. In the figures we plot the accuracy of classification (y-axis) against the number of single trial ERPs that were averaged to produce each of the ERP measures in a sample (each sample included 1000 ERP measures). Several aspects of the figures are noteworthy. First, for each of the pattern recognition techniques plotted, classification accuracy increased with increases in the number of trials per measure. This continued improvement in classification accuracy represents the increasing signal/noise ratio as additional single trials are averaged to produce each measure. Second, it is clear from the figures that the pattern recognition techniques improved at different rates and achieved different asymptotic levels of accuracy. For both of the subjects P3bp and P3area improved more quickly and achieved higher levels of performance than SWarea and P3cross. In fact, P3cross is not plotted for subject 2 because it never exceeded 50% classification accuracy.

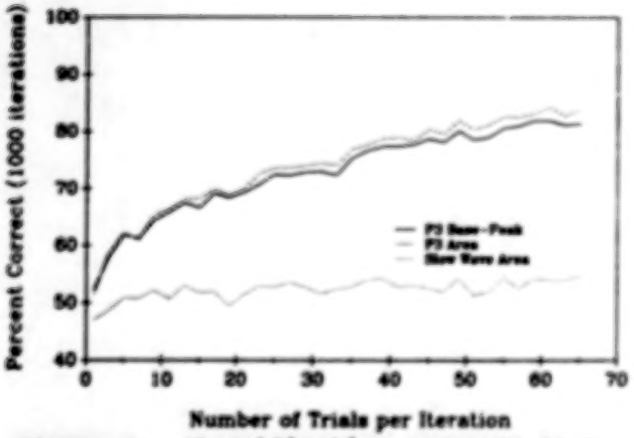


Figure 6. Classification accuracy as a function of the number of trials per measure for subject 2.

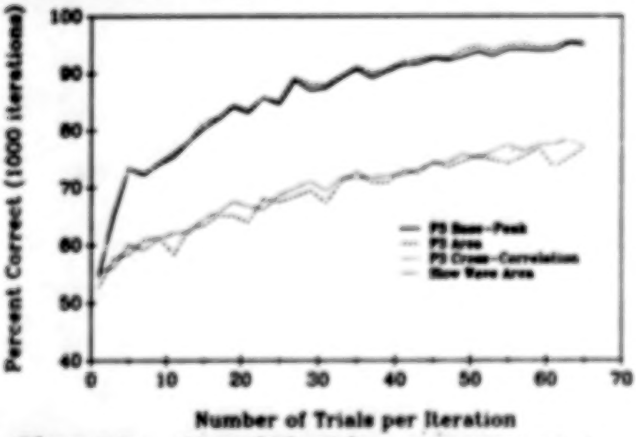


Figure 7. Classification accuracy as a function of the number of trials per measure for subject 3.

Third, for both P3bp and P3area there was a dramatic improvement in classification accuracy with the addition of the first five single trials followed by a more gradual improvement as additional trials were averaged. Finally, it is interesting to note that classification accuracy improved and reached different asymptotic levels for the two subjects.

SUMMARY AND CONCLUSIONS

The results of our investigation provide support for the utility of ERPs as real-time measures of mental workload. However, it is important to note that this support is both preliminary and tentative due to the small number of subjects, conditions, and pattern recognition techniques used in our study. The results are encouraging, however, and suggest a number of avenues for further exploration.

First, the differential efficiency of the pattern recognition techniques suggests that other techniques may offer improvements over the four that we have examined. In our study we used techniques that capitalized on the differences between only one component of the ERP (i.e. either P300 or Slow Wave amplitude). However, a number of other ERP components also appear to be sensitive to variations in mental workload (Horst et al., 1984; Kramer, 1987). Given that these components reflect changes in workload not indexed by P300 and Slow Wave amplitude, the use of multivariate techniques such as discriminant functions should improve the ability to discriminate among different levels of workload. It might also be possible to enhance discriminability by examining changes in the frequency spectra of EEG.

Second, previous examinations of the accuracy of single trial classifications of ERPs have suggested that the efficiency of different pattern recognition techniques is dependent on the characteristics of subject's waveforms (Farwell and Donchin, 1988). For example, base to peak measures tend to be most successful when the component of interest is sharpley defined while area measures are superior for wider components. Differences in the efficiency of P3cross and SWarea measures for our two subjects also appear to be due to differences in their waveforms. Thus, these analyses suggest that it might be useful to compile a set of heuristics that map waveform characteristics to pattern recognition techniques.

Third, it seems reasonable to suppose that the ability to discriminate among workload levels depends on the homogeneity within workload levels. In the present study we selected gauge samples in the LP/A3 condition irrespective of whether subjects were performing the arithmetic task (arithmetic tasks were presented with isi's of from 4 to 15 secs). Thus, our LP/A3 condition was actually a mixture of single and dual task trials. A comparison of the "dual task" trials in the LP/A3 condition with the LP condition should increase classification accuracy.

Fourth, while it is important to determine classification accuracy in the "best-case" situation it is also imperative that classification functions are derived for smaller differences in workload. We are currently examining the range of sensitivity of ERP measures to graded differences in workload. Finally, it is clear that classification accuracy can be improved by integrating psychophysiological and performance measures into predicative and descriptive equations. Therefore, it is necessary to determine how the relative sensitivity of different physiological and performance measures vary with changes in task structure and subject

state.

ACKNOWLEDGMENT

The research was supported by a grant from the Office of Naval Research (N00014-89-J-1493) with Dr. Harold Hawkins as technical monitor and by a grant from NASA Ames Research Center (NASA NAG-2-308) monitored by Dr. Sandra Hart.

REFERENCES

Brown, I. (1978). Dual-task methods of assessing workload. Ergonomics, 21, 221-224.

Donchin, E., Kramer, A., & Wickens, C. (1986). Applications of brain event-related potentials to problems in engineering psychology. In M. Coles, S. Porges & E. Donchin (Eds.), Psychophysiology: Systems, processes and applications. New York: Guilford.

Dornic, S. (1980). Language dominance, spare capacity and perceived effort in bilinguals. Ergonomics, 23, 369-377.

Farwell, L. & Donchin, E. (1988). Talking off the top of your head: toward a mental prothesis using event-related brain potentials. Electroencephalography and Clinical Neurophysiology, 70, 510-523.

Horst, R., Munson, R. & Ruchkin, D. (1984). Event related brain potential indices of workload in a single task paradigm. Proceedings of the Human Factors Society, 28th Annual Meeting. San Antonio, Texas.

Isreal, J., Chesney, J., Wickens, C. & Donchin, E. (1980). P300 and tracking difficulty: Evidence for multiple resources in dual task performance. Psychophysiology, 17, 259-273.

Jasper, H. (1958). The ten-twenty electrode system of the international federation. Electroencephalography and Clinical Neurophysiology, 10, 371-375.

Kahneman, D. (1973). Attention and effort. Englewood Cliffs, N.J.: Prentice Hall.

Knowles, W. (1963). Operator loading tasks. Human Factors, 5, 155-161.

Kramer, A. (1987). Event-related brain potentials. In A. Gale & B. Christie (Eds.), Psychophysiology and the electronic workplace. Sussex, England: Wiley.

Kramer, A., Sirevaag, E. & Braune, R. (1987). A psychophysiological assessment of operator workload during simulated flight missions. Human Factors, 29, 145-160.

Kramer, A., Wickens, C. & Donchin, E. (1985). The processing of stimulus attributes: Evidence for dual-task integrality. Journal of Experimental Psychology: Human Perception and Performance, 11, 393-408.

Natani, K. & Gomer, F. (1981). Electrocortical activity and operator workload: A comparison of changes in the electroencephalogram and in event related potentials. McDonnell Douglas Technical Report, MDC E2427.

Navon, D. & Gopher, D. (1979). On the economy of the human information processing system. Psychological Review, 86, 214-255.

Ogden, G., Levine, J. & Eisner, E. (1979). Measurement of workload by secondary tasks. Human Factors, 21, 529-548.

Rolfe, J. (1971). The secondary task as a measure of mental load. In W. Singleton, J. Fox & D. Witfield (Eds.), Measurements of man at work. London: Taylor and Francis.

Sanders, A. (1979). Some remarks on mental load. In N. Moray (Ed.), Mental workload: Its theory and measurement. New York: Plenum Press.

Sirevaag, E., Kramer, A., Coles, M. & Donchin, E. (1988). Resource reciprocity: An event-related brain potentials analysis. Acta Psychologica, 70, 77-97.

Wickens, C. (1979). Measures of workload, stress and secondary tasks. In N. Moray (Ed.), Mental workload: Its theory and measurement. New York: Plenum Press.

Wickens, C. (1980). The structure of attentional resources. In R. Nickerson & R. Pew (Eds.), Attention and Performance VIII. Hillsdale, N.J.: Erlbaum.

SHAPA:
AN INTERACTIVE SOFTWARE TOOL FOR PROTOCOL ANALYSIS APPLIED TO AIRC/EW COMMUNICATIONS AND WORKLOAD

Jeffrey M. James, Penelope M. Sanderson, and Karen S. Seidler
Department of Mechanical and Industrial Engineering,
University of Illinois at Urbana-Champaign
Urbana, IL 61801

ABSTRACT

As modern transport environments become increasingly complex, issues such as such as crew communication, interaction with automation, and workload management have become crucial. Much research is being focused on holistic aspects of social and cognitive behavior, such as the strategies used to handle workload, the flow of information, the scheduling of tasks, the verbal and non-verbal interactions between crew members. Traditional laboratory performance measures no longer sufficiently meet the needs of researchers addressing these issues. However observational techniques are better equipped to capture the type of data needed and to build models of the requisite level of sophistication. Presented here is SHAPA, an interactive software tool for performing both verbal and non-verbal protocol analysis. It has been developed with the idea of affording the researcher the closest possible degree of engagement with protocol data. The researcher can configure SHAPA to encode protocols using any theoretical framework or encoding vocabulary that is desired. SHAPA allows protocol analysis to be performed at any level of analysis, and it supplies a wide variety of tools for data aggregation, manipulation. The output generated by SHAPA can be used alone or in combination with other performance variables to get a rich picture of the influences on sequences of verbal or non-verbal behavior.

INTRODUCTION

Current Research Issues in Transport Environments

Today's cockpit environment is a challenging one. Crew members have to handle complex information that arrives through a variety of different channels: information management systems, visual displays, ATC, other crew members and so on. In addition, new on-board systems are continually being developed that need to be evaluated and refined in the context of piloting tasks.

Automation introduces a variety of issues. In many ways, it acts as an extra crew member. It is an entity that not only demands attention but is a source of information. As such, it contributes to the coordinative and information complexity in the cockpit. How do on-board computers affect how decisions are made and how information flows between crew members? Does automation decrease the amount of information overtly communicated, possibly resulting in misunderstanding? A computer may disrupt more accustomed exchanges between crew members by providing total or partial solutions to problems, or by interruption. Monitoring automated systems may itself produce workload. Alternatively, pilots may experience "underload", and feel removed from the basic piloting task. Does the operator trust the computer? Can the computer be used effectively to offload workload? Who takes responsibility for the allocation, and how does variation in workload between operators affect overall crew performance? These issues are being addressed by various researchers (Foushee, 1984; Wiener, 1985)

Because there are moments of distinct overload and quite long periods of underload, the management of workload becomes crucial. The individual must manage his or her own workload and, where relevant, do so in the context of how workload is allocated in a group of people. If a computer is present, its role in adding or taking away workload must be evaluated.

Hart (1982) has argued that overload and underload exist on a continuum. If overloaded, operators may defer or even shed tasks, and may choose tasks to perform on a different basis than usual. Hart argues that many tasks in a mission are discrete in nature and have "windows of opportunity" within which they can be completed. Given this, operators actually have a reasonable amount of discretion in how they organize their time, and may develop strategies for doing so. Workload then becomes a function not only of initial conditions such as task requirements, interface and operator resources, and operator experience, but also of how the operator assesses and reacts to the situation as it unfolds, and as he/she believes it will be in the future. Accordingly, the influence of conscious, strategic factors on workload and performance is now receiving a lot of attention.

Once it is accepted that the operator manages his or her workload strategically and creatively, research questions become less molecular and more molar. The determinants of workload will now lie in answers to questions such as: What does the operator know about the current situation? Is the operator happy with his or her level of performance? Does the operator have a particular strategy--or short-cut--for achieving a goal in the present situation? How does the operator manage an uneven level of workload? How does the operator prioritize tasks when they vary in terms of priority, time for completion, inherent cognitive or motor difficulty, and probability of being successfully completed within the time available? How will an operator's strategy change with time pressure?

It is clear that molecular information processing measures of performance such as reaction time and RMS error fail to answer such questions. New conceptual and methodological tools will be required. These tools will have to be sensitive to global, conscious, and deliberate aspects of behavior. In particular, it will be very important to be able to categorize behavior and detect patterns. Only on this basis can new models of workload be developed.

Observational Techniques

Many of the issues described above are not amenable to controlled experimentation, and traditional performance measures such as reaction time do not convey sufficient information to build the types of conceptual models needed. Observational techniques are far more suitable as they can provide ways of collecting data in a way that does not artificially

contain behavior and they provide various ways of analyzing data gathered in naturalistic environments. For example, non-verbal protocols can be collected of crew actions and gestures. In addition, there is much to be gained from eliciting conscious knowledge about strategies, tactics and other concerns from those we observe in such environments. Verbal protocols can be collected, where appropriate, to give an idea of the cognitive processes and strategies used. It is argued that many methods of data reduction used in ethology will be useful in analysing these two sources of data. Both these can be analysed alongside the more traditional performance measures.

There have been debates about the validity of verbal protocol data. Ericsson and Simon (1984) have argued that it should be possible to treat verbal data like any other sort of data. They argue that all data processing requires transformation from an initial observation to a form in which theories can be tested: "the cognitive processes that generate verbalizations are a subset of the cognitive processes that generate any kind of recordable response or behavior." (p. 9). As with any other data, researchers should be aware of the strengths and limitations of verbal data. For example, verbal data is most likely to be valid when a subject has been thinking about the task verbally and the information remains in short term memory.

If verbal data is to be a powerful tool in theory development and evaluation, then it is imperative that it can be subjected to various aggregation and analysis routines in order to achieve succinct representations of the information it contains. For example, it can be subjected to the types of data reduction techniques currently being used for observational data. Verbalization, after all, is one of the most important elements of the human behavioral repertoire, and can be used as a vehicle for organizing, directing and evaluating actions towards a goal.

The other type of protocol analysis is non-verbal protocol analysis. Non-verbal protocol analysis stretches to all types of observations of social behavior such as sequences of facial expressions, gestures, play behavior, and non-verbal communication (Schleer and Ekman, 1982). Such observational techniques are usually not nearly as intrusive, or potentially intrusive, as verbal protocol techniques, because the subject is not necessarily aware of being observed. Rigorous analytical techniques are seemingly better applied to the more objective, less self-conscious nature of non-verbal protocols. However, verbal and non-verbal data might be equally successfully analysed with ethological data reduction techniques.

When used together, verbal protocol analysis and non-verbal observational techniques are capable of capturing much of the richness of a situation being examined. However, both techniques involve detailed transcription and analysis, which are very time-consuming. One of the principal goals of the work discussed herein is to develop methodological tools that will allow such data to be analysed easily, and sound models to be developed quickly. The SHAPA environment, to be described, is a first step towards this. SHAPA is designed to afford researchers more direct engagement with a large, immensely rich, but hitherto relatively undigestible, body of data.

Purpose of SHAPA

Protocol analysis is notoriously difficult and time-consuming to perform. The time required for analysis of a protocol has been estimated to be an order of magnitude greater than the time required to actually record the protocol! This means that too often researchers use protocols as *anecdotal* support for theories or points being made, without building any kind of a statistical case on the basis of verbal data for the assertions that it is supporting. Any tools that can shorten this process and help verbal data be treated as any other sort of data, where appropriate, are invaluable.

It has been clear that what is needed is a highly general, interactive protocol analysis environment where the human encoder still makes the high-level judgments required, but the computer takes the burden for much of the "hack" work. SHAPA is a coordinated, interactive protocol analysis environment where researchers can encode a wide variety of data according to categories of their own choosing, in the context of a model or theory of their choosing. *Thus SHAPA is to protocol data what a spreadsheet program is to numerical data, or what a word processor is to text: it is intelligent about the sorts of things a researcher might want to do with verbal or non-verbal protocols, while being blind to particular domains, contexts, or theories.* SHAPA should allow a researcher to see more quickly the patterns in various types of sequential behavior. This should expedite data analysis and model development.

SHAPA allows the researcher to carry out the steps for protocol analysis suggested by Ericsson and Simon (1980, 1984) as well as others. Verbal statements, or segments of non-verbal behavior, can be encoded as *atomic formulae* from the predicate calculus. Predicate calculus is an AI-based language for expressing propositions and their relations in a standard format (see Charniak and McDermott, 1986). SHAPA does not assume that researchers always wish to use the entire predicate calculus to encode raw protocols. It merely uses the simpler of its conventions as an encoding syntax.

SHAPA provides a set of tools for identifying strategies and information flow from complex verbal and non-verbal protocols. At present, SHAPA best handles protocols from individual subjects, but we are embarking on a project to make it better handle protocols from multiple sources in parallel. SHAPA will continue to evolve as models of the impact of automation, crew communications and workload-management strategies develop. However it should, in its turn, accelerate the development of useful models in these areas. The use of such a tool speeds the development of useful models of strategic and interpersonal factors in cockpits, control rooms, or operations centers. It allows protocols to be analyzed at various levels of abstraction, allowing tests of different models of performance. Additionally, better models of workload can be built. We will be able better to identify and classify situations where performance will be threatened, providing predictive tools for designers of instrumentation and decision support systems.

PROTOCOL ENCODING USING SHAPA

The steps required for the more detailed encoding of protocols will now be enumerated, and the role of SHAPA described wherever relevant. This treatment deals mostly with the analysis of verbal data; however, it is easy to see how the technique could be extended to nonverbal data such as motor and performance data. In practice, the steps outlined here are seldom done in such a linear sequence: there is usually much back-tracking, adjustment and correction as protocol analysis gets under way. This is fully supported in SHAPA.

1. Task Analysis

Even before the protocol is taken, the researcher should understand the problem space of the particular task at hand. The important system states of the task environment should be identified, along with the important operators (actions). This is equivalent to identifying the "problem space" (Newell and Simon, 1972). The researcher should start to develop a

descriptive language for task states, perceptual states and operators. However, the development of the descriptive language can be helped by SHAPA.

It may only become clear what the syntax and vocabulary should be once the encoding has started. There can be a "bootstrapping" cycle here: the best descriptive language may only become obvious once the encoding begins. This is particularly true during the first few protocols in a set.

2. Transcription and Segmentation

The process of eliciting a verbal protocol from a subject will not be discussed here (the reader might consult the Appendix in Ericsson and Simon, 1984, for a short discussion of this). The stream of verbalization or behavior needs to be transcribed from audio or video tape and broken into segments for encoding. Researchers may also want to include time codes and symbolic representations of pauses in speech or behavior.

Segmentation involves breaking the stream of verbalization into sentences, clauses or phrases that express one idea and that can be encoded with one predicate (see 3. and 4.). Alternatively, segmentation can be performed where there are pauses in verbalization. The segments may be syntactically and/or semantically distinct, but must be sufficiently cognitively distinct to be encoded in a predicate.

3. Determine Encoding Vocabulary and Notation

Encoding vocabulary. As mentioned above, an encoding vocabulary, or descriptive language, needs to be developed. The basic idea is that when handling any sequence of behavior or verbalization, decisions have to be made about (1) what aspects of the situation are worthy of being noted, (2) what aspects of the situation must be distinguished from each other, and (3) what aspects are sufficiently similar to be classed together. When determining the encoding vocabulary, an important goal should be to reduce the variability in the natural language, or unconstrained behavior, to essential propositions which retain the semantics of the situation *in the context of the theoretical disposition of the encoders.*

SHAPA is relatively theoretically "agnostic" and has been designed to handle a wide variety of representations for problem-solving tasks. Researchers need to generate a working set of important distinctions they want to make about the behavior or utterances in the protocol record. Each of these distinctions can be qualified by what is, for all practical purposes, an infinite number of qualifiers or arguments. Obviously, however, an encoder wants to reduce a protocol to its "essentials", according to a theoretical viewpoint, so will choose qualifiers and arguments as parsimoniously as possible.

Encoding notation. Once the encoding vocabulary has been decided upon, there needs to be a standard, convenient notation for encoding--this will increase the reliability of the analysis. SHAPA uses something rather like the "atomic formula" of predicate calculus as a general encoding notation (Charniak and McDermott, 1986). The atomic formula consists of a *predicate* (also called an *operator*) and its *arguments* (also called its *terms*). Predicates can be verbs or nouns that represent the verbal or non-verbal activity that is of interest, or the propositional content of an utterance. The arguments qualify the predicate and provide details about the current situation. This fundamental representational syntax does not mean that SHAPA requires that thought or action be modeled according to the entire predicate calculus, even though it might sometimes be appropriate. It is simply a general and flexible way of representing the content of an utterance.

1. The predicate MONITOR might need to be qualified by (1) what is being monitored and (2) why it is being monitored.
2. The predicate COMMAND might need to be qualified by (1) who gives the command, (2) to whom it is directed, (3) the content of the command, and (4) the directness or indirectness of the command.
3. The predicate STATE might reflect comments about current system state, and include (1) the parameter being discussed and (2) its value.

The encoding vocabulary should handle as many of the verbalizations as possible, including queries, exclamations, and so on. Similarly, "place-holding arguments" should be chosen that indicate the syntax for the predicate arguments. Thus the *canonical* form of these predicates would be as follows:

1. MONITOR(<WHAT>,<WHY>)
2. COMMAND(<BY>,<TO>,<CONTENT>,<DIRECTNESS>)
3. STATE(<PARAMETER>,<VALUE>)

The place-holding arguments are surrounded by brackets to indicate that they have not yet been replaced by a *constant* based on the content of the raw protocol.

4. Perform Encoding.

This is where the SHAPA environment is used to its fullest extent. The SHAPA interface is similar to a full-screen editor, with certain constraints imposed by the nature of the raw protocol and protocol encoding files and the need to preserve their integrity. The encoder can move through the file much in the way one would in a word processor. SHAPA provides various screen format options for how the raw protocol and protocol encoding files can be displayed on the screen segment by segment. SHAPA's default layout alternates between lines of raw protocol and lines of encoding as shown in Figure 1.

Encoding individual segments. When the encoder wishes to encode a segment, he or she moves the cursor to the protocol encoding file line associated with that segment. He or she decides upon the appropriate predicate for the segment, and enters an abbreviated form of the predicate name (e.g., G for GOAL for C for COMMAND). SHAPA then displays the appropriate syntactical form of the predicate with the place-holding arguments--GOAL(<PARAMETER>,<VALUE>)--in the associated line of the protocol encoding file. The encoder then uses the cursor keys to move through the displayed predicate to replace place-holding arguments with standardized representations of the content of the protocol segment. For

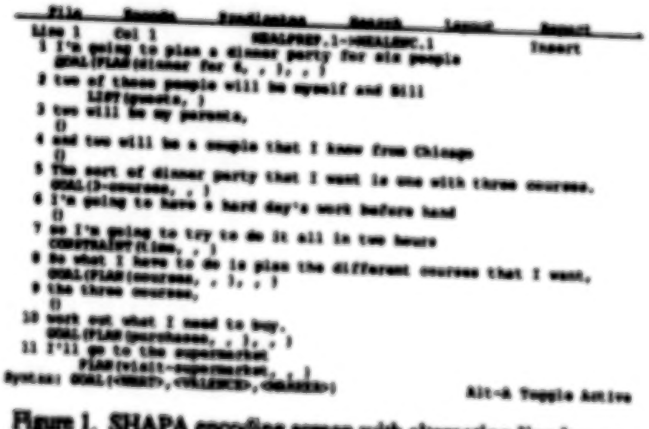


Figure 1. SHAPA encoding screen with alternating line layout

instance, the encoder might first replace <PARAMETER> with raw protocol-based content, leaving:

GOAL(flowrate,<VALUE>)

and then might replace <VALUE> with the following qualifier from the raw protocol:

GOAL(flowrate,increase).

The continual presence of the predicate as it is transformed from its canonical form to its encoded form ensures standardization which aids reliability of encoding.

It is important that the researcher decide in advance how much contextual information will be used to disambiguate unclear protocol segments, and how anaphoric references should be handled. Previous verbalizations or the logic of the task may make it clear what is being referred to. For the purest protocol analysis, where each verbalization is considered a single piece of data, replacing anaphoric references or missing referents with inferred referents is considered tantamount to "fudging the data". The level at which the use of context is appropriate is a judgment that needs to be made in advance by the researcher.

Aggregate encoding. So far the description has dealt with the encoding of individual segments only. However it is not necessary to provide an encoding for each protocol segment, and an encoder may actually want to encode a protocol at a higher level of abstraction or aggregation than this. Protocols can be encoded at multiple levels of abstraction: at a segment-by-segment level or at higher levels that might include "strategies", "episodes", "subroutines", or "phases". There may be habitual subroutines that are "run off" verbally with very little variability between instances. In this case, two or more segments might be encoded with a more global predicate.

SHAPA supports encoding at different levels of abstraction in two principal ways: (1) by providing the potential for inserting higher-level predicates on an encoding line of their own within a given protocol encoding file and (2) by providing the opportunity for encoding using multiple files.

Different researchers will inevitably approach a protocol from different theoretical viewpoints. These differences will probably be most markedly seen when encoding at a higher level. For instance, one researcher may wish to see how well the GOMS model accounts for the data (Card, Moran and Newell, 1983) and will tend to group predicates under higher-order predicates such as GOAL, OPERATOR, METHOD and SELECTION. Another researcher may wish to test whether a software change leads to the same operator function model (Mitchell and Miller, 1986) and might use previously determined categories to see if transitions between functions are still the same.

As mentioned, our use of the predicate notation does not imply that researchers must understand or adopt predicate calculus for their protocol analysis. Some of the fundamentals for doing so are present if this is desired. *The researcher brings his or her own theoretical disposition to the task of protocol analysis and the predicate-based protocol encoding file can be used in a wide variety of possible analyses.*

DATA MANIPULATION AND ANALYSES

Although a researcher can learn much from examining the encoded protocol, there is much that cannot be seen with "eyeballing". If verbal data are to be treated as any other sort of data, as Ericsson and Simon (1984) suggest, then verbal protocol analysis should be supported with tools that encourage exploration. Such tools should allow researchers to interact

```
SHAPA: Verbal Protocol Analysis
Encoding: MEALENC.1

JUDGE(<WHAT>,<WRT>,<AGE>,<VALENCE>,<MARKER>)
CONSTRAING(<WHAT>,<VALENCE>,<MARKER>)
GOAL(<WHAT>,<VALENCE>,<MARKER>)
PLAN(<WHAT>,<WHY>,<MARKER>)
LIST(<WHAT>,<MARKER>)
E-COMMENT(<WHAT>)
MEANS(<WHAT>,<MARKER>)

   1 GOAL(PLAN(dinner for 6, , ), , )
   2      LIST(guests, )
   5 GOAL(3-courses, , )
   7 CONSTRAINT(time, , )
   8 GOAL(PLAN(courses, , ), , )
  10 GOAL(PLAN(purchases, , ), , )
  11      PLAN(visit-supermarket, , )
  12 GOAL(PLAN(cooking-sequence, , ), , )
  14      CONSTRAINT(kitchen-help,present, )
  16 GOAL(3-courses, ,##)
  17      PLAN(MEANS(3-courses, ), , )
  18           CONSTRAINT(guests, , )
  19                LIST(CONSTRAINT(guests, , ), )
  35                     JUDGE(PLAN(meal, , ),GOAL(PLAN(meal, , )  ...
```

Figure 2. Report of (part of) encoding file without raw protocol

fluently with the verbal data, almost as one would with numbers in a statistical or spreadsheet program. Thus once the segments of a protocol have been encoded--at whatever level of aggregation--further analyses are available. In the following sections, statistical analyses and modeling tools available in SHAPA are described.

1. Indentation

One of the most important steps in understanding a problem solving strategy is understanding the flow of control of behavior. Superordinate goals, episodes or subroutines can be marked at their head with a suitable predicate. Subsequent activity that represents the means or strategies to reach those goals is subsumed under that heading. SHAPA incorporates a formatting facility where encoded segments can be grouped and indented to suggest flow of control.

The indentation procedure provides an interactive "visualization" tool for the researcher to experiment with different hypotheses about the flow of control. An indented file can be saved, providing information that can later be analyzed about the hierarchical structuring of verbalizations.

2. Filtering

After performing an encoding, a researcher will want to examine it various ways, either on the screen, in hard copy, or in a file dump. SHAPA allows the researcher to suppress different aspects of the raw protocol and protocol encoding files at output so as to have an output that highlights features of interest. Figure 2 shows part of a report listing predicate canonical forms and encoding lines without their raw protocol segments.

3. Report of Constants

Encoders will be interested in the size of the vocabulary they use to encode predicate arguments, and in how consistently they apply that vocabulary. At any point during encoding, SHAPA will quickly provide a report of the constants used under each "place-holding" argument of the predicates. For instance, in a aviation setting the predicate

GOAL(<PARAMETER>,<VALUE>)

may be examined. SHAPA will report its frequency of usage and the number of times GOAL was an embedded predicate. The constants used for <PARAMETER> may include *altitude, heading, direction, air speed* and *velocity*. On viewing this list the encoder may see that some constants are synonyms for others, as in the case of *heading* and *direction*.

4. Collection of Predicate Instances

For some applications, researchers will be interested in collecting together segments that have been encoded with the same predicate. For example, in a study of cockpit communication it is important to examine information inquiries among crew members in order to determine information flow. All segments requesting information could be encoded as INQUIRY statements. These statements needed to be collected together so that a graphical representation of the crew's flow of communication could be developed.

SHAPA will report selected encoding lines on the basis of the predicate name or on the basis of a particular constant used as one of the arguments for that predicate. Thus one might collect all INQUIRY statements regardless of the way their arguments have been encoded. Alternatively, one might collect all INQUIRY statements where the constant for the place-holding argument <SPEAKER> could be, for example, captain. This would allow one to examine all inquiries made by the captain of the aircraft.

5. Cross-Reliability

In the context of verbal protocol analysis, reliability is the tendency for one encoding of a protocol to be similar to another, whether they be separate encodings by one person or by different people. Verbal protocol analysis cannot have any pretensions to validity unless at some level it is reliable. Good agreement between encoders suggests a clear, unambiguous protocol with respect to the encoding vocabulary chosen, but does not guarantee that what has been said bears a faithful relationship to the mental processes involved in the task in question. Thus reliability does not guarantee validity.

SHAPA provides a reliability cross-check between the predicates used in two encodings of the one raw protocol file. It can compare encodings performed with the same set of predicates or two different sets of predicates. When comparing two files, SHAPA sets up a matrix that has the predicates used in the first encoding as the rows, and the predicates used in the second encoding as the columns. Each encoded segment is represented as a tally in the appropriate cell of the matrix. When the rows and columns represent the same set of predicates, then two encodings with good reliability would reveal a large number of observations on the left diagonal. With this matrix approach, predicates resulting in high and low reliability can easily be distinguished. Another feature of this approach is that the predicates on the rows and columns need not be the same. The matrix-based reliability cross-check allows a researcher to see which predicates might be considered to refer to the same thing, and which predicates have only partially overlapping meanings.

6. Transition Matrices

An important way of capturing patterns in the protocol record is to look at the sequencing of behavior. Transition frequency matrices can be constructed from which a variety of analyses can be performed (van Hooff, 1982). Transition analyses help to

First Order Transition Matrix

Predecessor	Successor						
	J	C	G	P	L	EC	W
JUDGE	5	2	1	5	1	.	2
CONSTRAINT	1	1	2	1	1	.	1
GOAL	1	2	1	5	1	.	.
PLAN	5	1	2	5	5	.	1
LIST	2	.	1	4	.	1	.
S-CONCEPT	.	.	1	.	.	.	.
MEANS	2	.	1	.	1	.	.

Figure 3. First order transition matrix

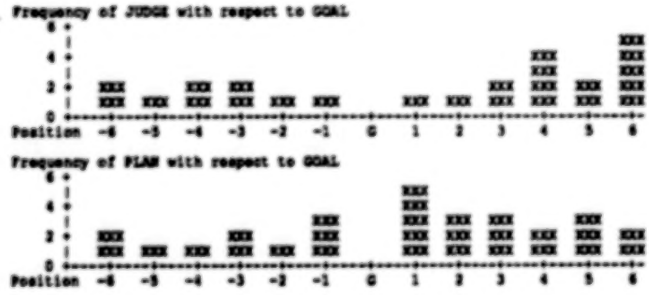


Figure 4. Lag sequential analysis

determine whether a verbal segment or piece of behavior is generally strongly influenced by the behavior before it; that is, whether there are dependencies in the data matrix. Such analyses allow the researcher to detect subsystems of verbalization or behavior that serve a specific function.

The analysis may thus reveal habitual or stereotyped patterns of behavior, such as GOAL-->STATUS-->PLAN. Such patterns may indicate elemental functional *nexi* such as planning a course of action or anticipating future activity. It is then up to the researcher to determine how much the regularity of the pattern is due to environmental constraints and how much to human preferences and strategies, and to draw conclusions appropriate to the hypotheses being tested. A first order transition matrix generated by SHAPA is shown in Figure 3. SHAPA also provides second and third order transition matrices which are potentially one and two orders of magnitude larger.

7. Lag Sequential Analysis

In some domains, whether with verbal or non-verbal data, there may be a tendency for one type of utterance or activity to precede or follow another at a certain remove. For instance, in crew communication a response may tend always to come between two and four statements following an inquiry. This general tendency might be missed by transitional matrix analyses, where strict sequences of predicates are used. Lag sequential analysis allows these more vague patterns and dependencies in the sequence of protocol segments to be discerned (Douglas and Tweed, 1979). Figure 4 contains two graphs of a lag sequential analysis report generated by SHAPA. Separate graphs are generated for each available predicate.

8. Frequency of Cycles

The final analysis has been adopted from Fisher (1988), and is similar to various ethological techniques for finding regularities in behavior sequences. "Fisher's Cycles", as we will call them, provide a report of actually occurring sequences of predicates rather than formally defined sequences as in transition analyses. It is a powerful heuristic for identifying the patterns in the data that relies upon human rather than machine pattern recognition ability. It also avoids the intensive computation often needed to identify and compare patterns.

More details about SHAPA can be found in Sanderson, James, and Seidler (1989).

CONCLUSION

As a piece of software, SHAPA is in the cognitive engineering tradition of exploiting human pattern-recognition abilities to amplify the user's "intelligence" (Norman and Draper, 1986; Woods and Roth, 1988). SHAPA is designed to facilitate the initial determination of appropriate vocabularies for encoding and to speed up the encoding and analysis of a series of files once an encoding scheme has been established. In this way, researchers can encode a greater number of protocols and generate measures of verbal or non-verbal behavior that can more quickly become amenable to conventional inferential statistics, or other pattern-recognition techniques. SHAPA is designed to engender a feeling of *direct engagement* with the protocol data and to make the manipulation of data as direct as possible.

With experience in analyzing verbal protocols, the researcher will have educated himself or herself in what to look for in a protocol--the patterns of verbalizations or actions and the evidence for theoretically important distinctions. Future protocols should then be much easier to analyze because the patterns should be easier to discern. Ideally, at this point the researcher should have the conceptual tools to classify subjects' conscious strategies on the basis of a smaller, more diagnostic sample of behavior. One may finally reach the point where a well-structured questionnaire, interview or behavioral test allows the subject to classify themselves reliably with respect to the categories important for the research at hand (see Sanderson, 1989, for an example on a small scale).

Protocol analysis and observational techniques support the kind of conceptual model-building that is sufficient to support design decisions. They often suggest hypotheses that might be tested in further observation or in more controlled experimental manipulations. One of the byproducts of the proposed work is that the conceptual advances offered by model building in the SHAPA encourage more focussed observational techniques, or experimental 'spin-offs' that can be performed in more controlled environments. In other words, SHAPA should help researchers learn what to look for, and how to measure it.

As further requirements of protocol analysis come to light, SHAPA will be developed to incorporate them as far as possible. In the future, SHAPA will be developed on the Macintosh so as to include more "visualization aids" where researchers can explore different ways of graphically representing protocol data to show patterns in their data. SHAPA will also be streamlined so that it more comfortably handles handle protocol data from multiple interacting agents, such as groups of people or a person interacting with a computer.

Major developments of SHAPA are currently being determined. Development of SHAPA will include integrating time into the protocol analysis. Time stamping of encoded lines will allow coordination with video protocols through a VCR interface as well as with continuous and discrete status variables taken from the environment. Features and analyses for integrating multiple encodings of the same protocol will be considered. Analysis of results across protocols will allow for a more holistic approach once simple protocol analysis through SHAPA has established. Further filtering and isolation of status variables, predicates, and arguments, and representation of data patterns in graphical form, will allow for greater visualization of the encoding.

Developing such a coordinated data analysis environment will be a large job, but not a conceptually difficult one once the basic needs and constraints had been identified. In the meantime, we feel that SHAPA represents a significant step towards solving some of the practical problems of performing protocol analysis, such as swift access to the data and speed of manipulation. It also enriches the types of conclusions that can be drawn by importing analyses from the non-verbal domain.

REFERENCES

Card, S., Moran, T., and Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Erlbaum.

Charniak, E. and McDermott, D. (1986). *Introduction to Artificial Intelligence*, Reading, MA: Addison-Wesley.

Douglas, J.M. and Tweed, R.L. (1979). Analysing the patterning of a sequence of discrete behavioral events. *Animal Behavior*, 27, 1236-1252.

Ericsson, K.A. and Simon, H.A. (1980). Verbal reports as data. *Psychological Review*, 87, 215-251.

Ericsson, K.A. and Simon, H.A. (1984). *Protocol Analysis*. Cambridge, MA: MIT Press.

Fisher, C. (1988). Advancing the study of programming with computer-aided protocol analysis. In G. Olson, E. Soloway, and S. Sheppard (Eds), *Empirical Studies of Programmers, 1987 Workshop*. Norwood, NJ: Ablex Publishing Corporation.

Foushee, H. C. (1984). Dyads and Triads at 35,000 feet: Factors affecting group process and air crew performance. *American Psychologist*, 39, 885-893

Hart, S. (1988). Presentation given to NASA contractees at Human Factors Society Meeting, Anaheim, CA. October 28, 1988

Mitchell, C. and Miller, R.A. (1986). A discrete control model of operator function: A methodology for information display design. *IEEE Transactions on Systems, Man and Cybernetics, SMC-16*, 343-357.

Newell, A. and Simon, H.A. (1972). *Human Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall.

Norman, D.A. and Draper, S.W. (1986). *User-Centred System Design: New Perspectives on Human-Computer Interaction*. Hillsdale, NJ: Erlbaum.

Sanderson, P. M. (1989). Verbalizable knowledge and skilled task performance: Association, dissociation, and mental models. *Journal of Experimental Psychology: Learning, Memory and Cognition, 15*, 729-747.

Sanderson, P.M., James, J. and Seidler, K. (1989). *SHAPA: A software environment for verbal and nonverbal protocol analysis*. (Tech. Report EPRL-89-09). Urbana-Champaign: Engineering Psychology Research Laboratory, University of Illinois.

Van Hooff, J.A.R.A.M. (1982). Categories and sequences of behavior: methods of description and analysis. In K.R. Scherer and P. Ekman (Eds.) *Handbook of Methods in Nonverbal Behavior Research*. Cambridge: Cambridge University Press.

Wiener, E. L. (1985). Beyond the Sterile Cockpit. *Human Factors*, 27, 75-90.

Woods, D.D. and Roth, E. (1988). Cognitive Systems Engineering. In M. Helander (Ed.) *Handbook of Human-Computer Interaction*. New York: Wiley.

USEFULNESS OF HEART MEASURES IN FLIGHT SIMULATION

Randall L. Harris, Sr.; Gregory A. Bonadies; and J. Raymond Comstock, Jr.
NASA Langley Research Center
Mail Stop 152E
Hampton, VA 23665-5225

ABSTRACT

The results of three studies performed at the NASA Langley Research Center are presented to indicate the areas in which heart measures are useful for detecting differences in the workload state of subjects. Tasks that involve the arousal of the sympathetic nervous system, such as landing approaches, were excellent candidates for the use of average heart-rate and/or the increase in heart-rate during a task. The latter of these two measures was the better parameter because it removed the effects of diurnal variations in heart-rate and some of the intersubject variability. Tasks which differ in the amount of mental resources required are excellent candidates for heart-rate variability measures. Heart-rate variability measures based upon power spectral density techniques were responsive to the changing task demands of landing approach tasks, approach guidance options, and 2 versus 20 second interstimulus-intervals of a monitoring task. Heart-rate variability measures were especially sensitive to time-on-task when the task was characterized by minimal novelty, complexity, and uncertainty (i.e., heart-rate variability increases as a function of the subjects "boredom").

INTRODUCTION

The Human Engineering (HEM) Group at Langley Research Center (LaRC) utilizes physiological measures to characterize the impact of various flight management displays and/or controls upon the pilot's mental state. Heart-rate parameters are being investigated for use in such display and control evaluations. Instantaneous and average heart-rate measures have been used quite successfully by other researchers in several flight studies to evaluate the effect of the steepness of an Instrument Landing System (ILS) flight path on pilot heart-rate (ref. 1). In addition, positive correlations between heart-rate and subjective estimates of workload has been reported (ref. 2). However, other types of tasks have not affected the heart-rate as systematically (ref. 3). In an effort to determine the advantages and limitations of these measures in the flight deck environment, and to establish protocols and guidelines for their application, the HEM Group has undertaken a series of piloted studies. This paper presents data from three studies and examines the usefulness of two classes of EKG measures, heart-rate and heart-variability. The discussion provides guidelines for choosing measures judiciously, as well as for evaluating circumstances for which particular choices of cardiac response measures are germane.

RATIONALE FOR USE OF HEART MEASURES

Measures of changes in average heart-rate reflect the synergistic action of sympathetic and parasympathetic nervous systems on the cardiovascular control system (ref. 4) and have been considered an index of general arousal (ref. 5). Therefore, it would be expected that any task having an affect on the autonomic nervous system would affect heart-rate. In addition, tasks involving physical activity will affect heart-rate.

A number of measures of heart-rate variability may also be derived from the electrocardiogram (EKG) signal. Unfortunately, the term "variability" can refer to any one of several methods of analysis of heart-rate variability. Such methods include statistical variability measures, such as the standard deviation, which may be derived over a number of inter-beat-intervals (IBIs) or a selected time interval, and spectral frequency analyses of the IBIs.

Spectral analysis of the IBI data has shown promise in the assessment of mental workload (ref. 6). From the spectral analysis the amount of power in a given frequency band may be calculated. Power in a frequency band from .05 to .15 Hz has been shown to reflect the action of neural processes on arterial blood pressure regulation mechanisms (ref. 7). Changes in power in this frequency band also show sensitivity to changes in task demands. The amount of heart-rate variability has been shown, under

laboratory conditions, to decrease with increasing task difficulty. However, interpretation of phasic changes in the cardiac interval signal as a function of task demands is complicated by non-linearity in the relevant physiological control systems (ref. 7). This non-linearity means that a linear change in the heart-rate variability index should not be expected with a linear change in task demand. Despite this limitation, frequency spectral analysis of the signal is considered promising as an estimate of aircrew mental workload (ref. 8). Mental tasks, which require physical responses to implement a decision, reportedly produce changes in heart-rate variability (ref. 9). Therefore, heart-rate variability measures appear particularly appropriate for use in evaluating pilots' mental state during flight simulation tasks.

PILOTED STUDIES

The first two studies were performed in Langley's Visual/Motion Simulator (VMS). The VMS has a virtual image system to show the visual image taken of a model terrain board scene around a runway. The simulation tasks consisted of performing landing approaches. Reference 10 provides a more complete description of the simulation facility. The exterior of the simulator is shown in figure 1 and the interior is shown in figure 2. The cockpit was configured to simulate a jet transport. The experimenter performed the functions of the first officer in the right seat. Cardiac IBI data were collected for the pilots during flight simulation test runs. Each simulation run was preceded by a baseline period of several minutes duration, during which cardiac IBI data were collected. These baseline data were collected between simulation runs and may not represent truly "resting" conditions, because the subjects were either engaging in subjective workload assessment in the form of a verbal report or discussing conceptual aspects of the study or procedural aspects of the simulation task with the experimenter.



Figure 1. VISUAL MOTION BASE SIMULATOR



Figure 2. VISUAL MOTION BASE INSTRUMENT PANEL

The first study utilized five airline pilots and one NASA test pilot. Each completed 36 landing approach runs in the VMS for evaluation tests in flight along complex area navigation paths within the Microwave Landing System (MLS) signal environment. The primary independent variables were the type of flight path: ILS, RIVER, or HOOK. The ground track of the ILS path consisted of two straight lines, while that of the RIVER looked like the windings of a river with a short straight final to a landing, and that of the HOOK looked like a question mark. The task of flying the HOOK path was made even more difficult by removing the predictive elements from the guidance displays leaving the pilot to fly with reference to raw MLS data. Choice of these particular paths and control guidance configurations was based upon a prior set of tests that had shown these paths to have different Subjective Workload Assessment Techniques (SWAT) ratings (ILS was 14.8, RIVER was 31.8, and HOOK was 54.4).

The second study (a stereopsis/display format study) was also performed in the VMS, and utilized six United States Air Force transport pilots. Each pilot completed 24 landing approach simulation runs to evaluate the use of a perspective, stereo 3-D, path-in-the-sky display. The two main factors in the study were stereo versus non-stereo presentation and the type of pathway symbology. The pathway symbols were goalpost, monorail, and triangle-based monorail. Figure 3 shows a pilot in the simulator using goggles which alternatively transmit a left- and right-eye view, thereby providing stereopsis-type depth cues in the stereo version of the perspective display.



Figure 3 STEREOGRAPHIC EQUIPMENT IN THE VISUAL MOTION SIMULATOR

The third study was a laboratory vigilance study concerned with physiological and performance assessment of subjects in a task underload scenario. Some preliminary data from this study were reported previously (ref. 11). In this study, subjects monitored a CRT display containing a schematic of a jet aircraft engine. The subjects were instructed to take corrective action by depressing a particular key on the keyboard anytime one of five areas turned red, thereby restoring the display to its normal condition. The engine schematic and key layout are shown in figure 4. The task lasted for 1 hour. The period between stimuli was either a fixed interval of 6 seconds or alternated between 2- and 20-second intervals with 5 minutes at each level.

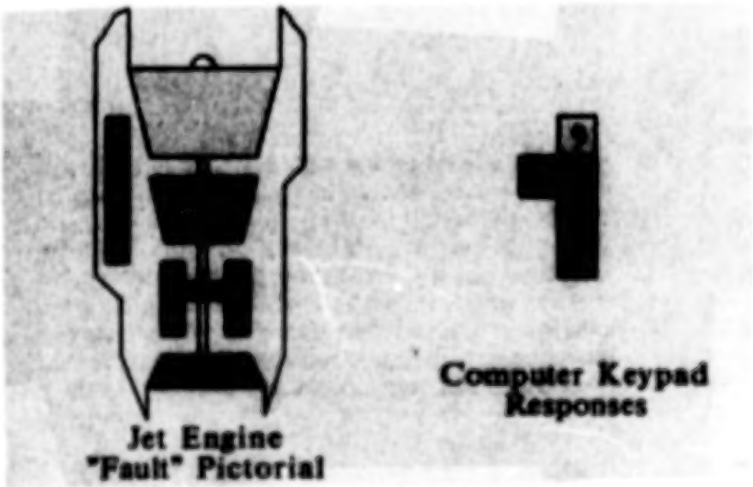


Figure 4. FAULT ACKNOWLEDGEMENT TASK

CARDIAC RESPONSE MEASURES

The electrocardiogram (EKG) signal (figure 5), from which the IBI, heart-rate, and heart-rate variability measures were derived, was obtained through active electrodes attached to the top of the sternum and the lower left rib cage. A reference electrode was attached to the left ankle. The EKG signal was fed through an optically isolated bioamplifier and routed to either signal processing equipment or magnetic tape depending on the requirements of the study. A level-sensitive Schmitt-trigger was used to determine the EKG IBI by timing and recording the duration of the intervals between successive cardiac contraction signals as they cross a preset level of the Schmitt-trigger hardware (figure 5). Although this technique does not detect the time of the actual peak of the "cardiac signal" the resulting error is much less than 3 milliseconds, which is of equal or greater precision compared to the timing of most other techniques in common usage. The series of inter-beat-intervals (IBI's), in units of milliseconds, is preserved in a file for later processing and analysis. The IBI data was used to calculate the average heart-rate (AHR).



Figure 5. INTER-BEAT-INTERVAL DETECTION SCHEME

The spectral analysis measure of heart-rate variability was obtained from a Fourier analysis of the IBI data. Software algorithms convert the IBI data sequence into an equal-spaced time series sampled at 4 Hz (cardiotachogram step-function). A low-pass digital filtering algorithm (ref. 12) was then employed to filter the sampled cardiotachogram. Fourier analysis of the filtered cardiac event sequence yields a spectrum of frequency components in the range 0.0 Hz to approximately 0.5 Hz (the maximum frequency is actually limited by the heart-rate).

Two main summary measures of heart-rate variability are derived from the frequency spectral analysis: (1) total heart-rate variability (THRV), which is the total area under the power versus frequency (spectral density) curve (0.0 to 0.5 Hz) and (2) blood pressure component of heart-rate variability (BPHRV), which is the area under the power spectral density curve in the frequency range of 0.05 to 0.15 Hz. As noted previously, changes in this frequency band have been shown to reflect the action of arterial blood pressure regulation mechanisms (ref. 7).

RESULTS

MLS Approach Paths

The first study compared heart measures for three different MLS approach paths (remember that the HOOK approach also involved the pilot's use of raw "path deviation" data instead of command guidance data). Table 1 presents the means and standard deviations of four heart measures (Total heart-rate variability - THRV, band pass heart-rate variability - BPHRV, Average heart-rate - AHR, and Average heart-rate minus Baseline heart-rate - AHR-BL) for the three approach paths. AHR and AHR-BL values were greatest for the HOOK approach and the THRV and BPHRV were lowest for the HOOK approach. The Analysis of Variance shows a significant difference in the AHR-BL parameter for the three approach paths ($p < .034$). The increase in heart-rate was sensitive to the workload change involved in the different paths (straight path of ILS and the multiple-curve path of the RIVER approach) as well as the increased mental stress of the combination of path without command guidance information of the HOOK approach. Even though the differences in the variability measures do not show statistical significance, the trend in the mean values follow the subjective workload ratings. Highest heart-rate variability with the lowest rated workload task and vice versa.

Table 1. Heart Measures

	THRV ms²/ms	BPHRV ms²/ms	AHR BPM	AHR-BL BPM ●
	Mean (Standard Deviation)			
ILS	60.43 (22.44)	22.81 (8.10)	72.24 (6.00)	0.70 (1.80)
RIVER	55.17 (20.22)	20.65 (6.65)	73.51 (5.92)	2.36 (1.81)
HOOK	53.70 (25.23)	19.73 (9.18)	74.29 (7.24)	3.18 (3.00)

● $p < .034$

Stereopsis/Display Format

The stereopsis/display format piloted simulation study involved the pilots' use of perspective, path-in-the-sky displays for curved, decelerating, descending approach-to-landing under turbulent wind conditions. Figure 6 shows the heart-rate of one pilot making a landing approach. Characteristic features of this heart-rate time history show a constant heart-rate at the beginning of the approach until the introduction of winds, and followed by flight maneuvers involving a reduction in airspeed, a curved path to aligning up with the runway, and a final straight segment leading to a touchdown and subsequent stopping on the runway. Table 2 lists the difference in heart-rate at touchdown and the heart-rate at the beginning of the landing approach (AHR-BL) for the two experimental manipulations of stereopsis and displayed pathway. This measure is used in lieu of the heart-rate at touchdown, because it reduces some of the between-subject variability. Unfortunately, it does not eliminate that variability entirely. On the average, the AHR-BL was less with stereo than without stereo. There was a significant difference in the AHR-BL measure for the stereopsis factor ($p < .027$), with a significantly smaller change in AHR-BL with the use of stereo. This shows that the level of arousal or stress was decreased through the use of the stereo display. Differences in AHR-BL between the pathway symbology types were not significant. Pilot comments were likewise ambivalent concerning the effects of the pathway symbology.



Figure 6. HEART RATE RESPONSES

Table 2. AHR-BL Heart Measures

Stereopsis ●	On	Off
	9.67 (5.87)	11.44 (6.22)

Pathway	Goalpost	Monorail Triangle-Base	Simple Monorail
	10.08 (5.52)	12.66 (8.77)	8.93 (5.94)

● $p < .027$

Stereopsis effects on the variability measures, THRV and BPHRV, are shown in figure 7 for overlapping time periods during the landing approach. The data show a consistent decrease in the variability measures from the first to the last of the run. This decrease in the heart rate variability parallels the increases in task difficulty of the landing approaches brought on by winds and required changes in aircraft state as the airplane proceeded toward a landing. While there were no statistically significant differences in the THRV or BPHRV measures for the stereopsis effects, there were differences during the landing approach.

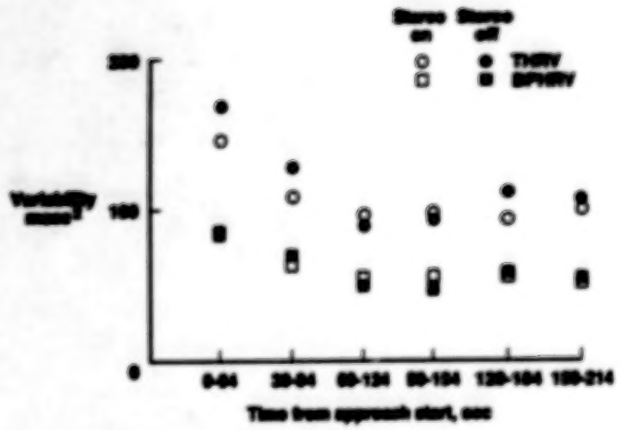


Figure 7. STEREOPSIS EFFECT ON HEART RATE VARIABILITY

Pathway effects on the variability measures, THRV and BPHRV, are shown in figure 8 for overlapping time periods of the landing approach. These data also show a consistent decrease in the measures from the first three segments with a very slight increase in the last segments. Consistent with pilot comments, the statistical analysis showed no significant effect of the type of pathway upon either THRV or BPHRV. A consistent trend for the first three time segments of the data was that the heart-rate variability for the triangle pathway was consistently lower than the other pathways. This would indicate a higher mental workload associated with that pathway symbology during those phases of the flight.

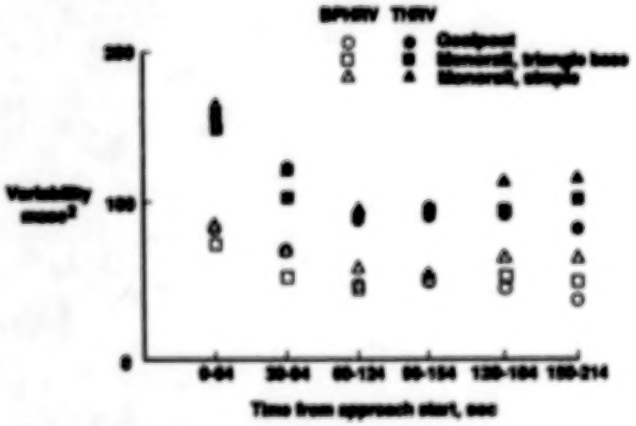


Figure 8 PATHWAY EFFECTS ON HEART-RATE VARIABILITY

Task Underload

For the task underload study, the subjects spent 1 hour monitoring a display to respond to fault indications with the press of a button. Three heart-rate parameters were derived (AHR-BL was omitted) for consecutive 5-minute blocks of a 1-hour task. Data were collected for two different inter-stimulus-interval (ISI) conditions, constant 6 seconds and alternating 5-minute blocks of 2 and 20 seconds.

Figures 9 thru 11 show AHR, THRV, and BPHRV parameters for each 5-minute block of the two ISI schedules. The ANOVA analysis of these data are presented in Table 3. The two variability parameters, THRV and BPHRV, statistically differentiated the block factor.

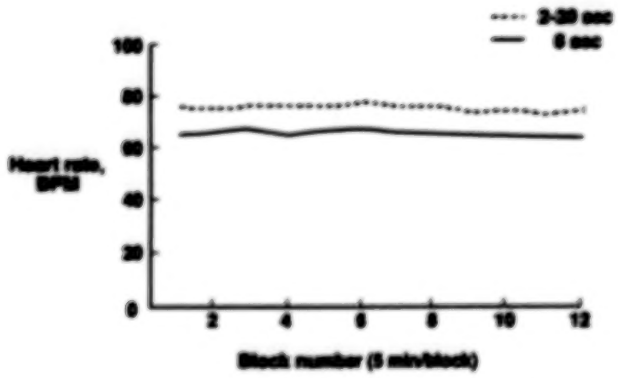


Figure 9. AVERAGE HEART RATE VERSUS TIME ON TASK

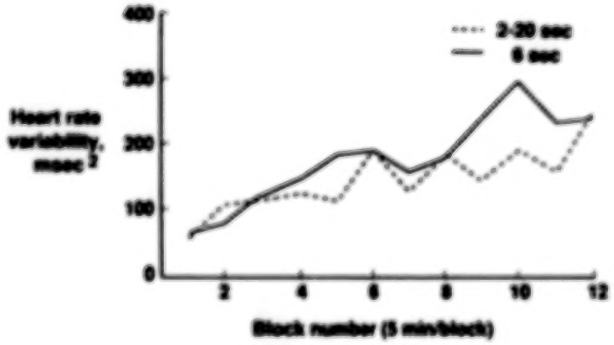


Figure 10. HEART RATE VARIABILITY VERSUS TIME ON TASK

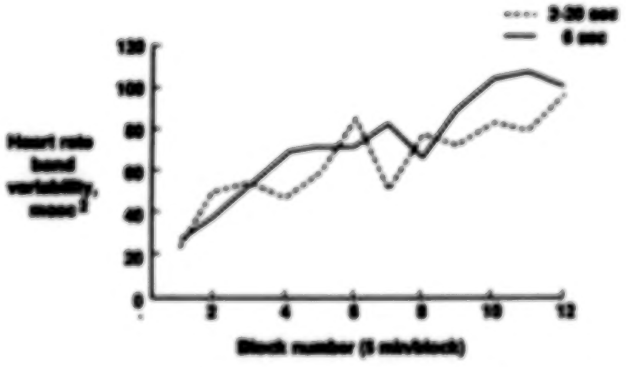


Figure 11. HEART RATE VARIABILITY (.05 - .15 HZ BAND) VERSUS TIME ON TASK

Generally, THRV and BPHRV measures were greater with increasing time on task (block number). The ISI alternating condition of 2-20 seconds was marginally significant for the THRV parameter. The differences in the mental workload of the 2-20 seconds ISI blocks is evident in the changes in the heart-rate variability measure but not in the heart-rate measure. Since there were no changes in the task demands with block number (except alternately for the 2-20 seconds ISI) the largest changes in heart-rate variability was that of block number (time-on-task). This indicated something about the mental state of the subjects as a function of time on the task and suggests that the mental resources devoted to the task were decreasing. This effect was shown by the increases in the heart-rate variability measures with block number.

Table 3. ANOVA significance levels

Factor	THRV	BPHRV	AHR
Blocks(6 SEC)	<.001	<.001	ns●
Blocks(2-20 SEC)	<.043	<.019	<.053
ISI (2-20 SEC)	<.073	ns●	ns●

● ns = not significant (p > .10)

CONCLUSIONS

Taken together the results of these three studies can be used to indicate the areas in which heart measures are useful in measuring differences in the workload state of subjects.

Tasks that involve the arousal of the sympathetic nervous system, such as the landing approaches used in the first two experiments, are excellent candidates for the use of the two heart-rate measures: (1) average heart-rate (AHR) and (2) increase in heart-rate during a task (AHR-BL). Of these two measures, AHR-BL is the better parameter, because it removes the diurnal variations in heart-rate, and it tends to reduce some of the inter-subject variability. In order to measure the AHR-BL parameter some provision has to be made for taking a baseline measurement, either preceding the data run, as in the case of the first experiment, or by designing the data run to include a non-demanding period (a reference heart-rate), as in the second experiment. This latter technique has the advantage of controlling the activity of the subject while taking the baseline measure.

In addition to heart-rate, heart-rate variability measures are also very responsive during the landing approach tasks. Although heart-rate increases during the landing approach and heart-rate variability decreases during the landing approach, both sets of measures reflect the increasing task demands of the landing maneuver.

Heart-rate variability measures show sensitivity to some task demand changes (2-20 ISI, MLS approach guidance options, and stereopsis effects). In addition, increases in heart-rate variability have also been used successfully in the evaluation of changes in the boredom level during a test run of tasks which can be characterized by minimal novelty, complexity, and uncertainty (i.e., heart-rate variability increases as a function of "boredom").

Neither one of the two heart-rate variability measures seem to be the more sensitive measure. In one of the tests, THRV was the more sensitive parameter, in another test, the BPHRV was the more sensitive parameter, and in a third test, both THRV and BPHRV were sensitive measures. At this point in time, it cannot be predicted which of the two variability measures will be the more sensitive. Either parameter can be derived in the Fourier analysis of the data. The researcher will have to derive and analyze both parameters.

REFERENCES

1. Roscoe, A. H., "Pilot Workload During Steep Gradient Approaches," Technical Memorandum No. TM FS 78, Royal Aircraft Establishment, Flight Systems Department, Farnsborough, England, 1978.

2. Roscoe, A. H., "Heart-Rate Changes in Test Pilots", The Study of Heart-Rate Variability, Ed. Kitney, R. I., and Rompelman, O., Oxford: Clarendon Press, 1980.

3. Mobbs, R. F., David, G. C., and Thomas, J. M., "An Evaluation of the Use of Heart-Rate Irregularity as a Measure of Mental Workload in the Steel Industry," BISRA, OR/HF/25/71, British Steel Corporation, London, England, August, 1971.

4. Linden, R. J. "Sympathetic and Parasympathetic Control of the Heart," Pychophysiology of Cardiovascular Control, Ed. Orlebeke, J. F., Mulder, G., and Van Doornen, L. J. P., Plenum Press, New York, 1985.

5. Duffy, E., Activation and Behavior, John Wiley & Sons, New York, 1962.

6. Mulder, G. "Sinusarrhythmia and Mental Work Load," Mental Workload Theory and Measurement, Ed. Moray, N., Plenum Press, New York, 1979.

7. Sayers, B. McA. "Physiological Consequences of Informational Load and Overload," Research in Psychophysiology, Venables, P. H., and Christie, M. J., John Wiley & Sons, New York, 1975.

8. Wierwille, W. W., "Physiological Measures of Aircrew Mental Workload," Human Factors, Vol. 21, No.5, October, 1979, pp. 575-593.

9. Hyndman, B. W., and Gregory, J. R. "Spectral Analysis of Sinus Arrhythmia during Mental Loading." Ergonomics, Vol. 18 No. 3, May, 1975, pp. 255- 270.

10. Parrish, Russell V., and Bowles, Roland L. "Motion/Visual Cueing Requirements for Vortex Encounters During Simulated Transport Visual Approach and Landing." NASA TP-2136, April, 1983.

11. Comstock, J. Raymond, Jr., Harris, Randall L, Sr., and Pope Alan T. "Physiological Assessment of Task Underload." NASA CP-3019, July, 1988.

12. Graham, Ronald J., "Determination and Analysis of Numerical Smoothing Weights." NASA TR R-179, December, 1963.

SPACE ENVIRONMENTS AND THEIR EFFECTS ON SPACE AUTOMATION AND ROBOTICS

by

Henry B. Garrett
The Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA. 91109

ABSTRACT

Automated and robotic systems will be exposed to a variety of environmental anomalies as a result of adverse interactions with the space environment. As an example, the coupling of electrical transients into control systems, due to EMI from plasma interactions and solar array arcing, may cause spurious commands that could be difficult to detect and correct in time to prevent damage during critical operations. Spacecraft glow and space debris could introduce false imaging information into optical sensor systems. The presentation provides a brief overview of the primary environments (plasma, neutral atmosphere, magnetic and electric fields, and solid particulates) that cause such adverse interactions. The descriptions, while brief, are intended to provide a basis for the other papers presented at this conference which detail the key interactions with automated and robotic systems. Given the growing complexity and sensitivity of automated and robotic space systems, an understanding of adverse space environments will be crucial to mitigating their effects.

INTRODUCTION

The space environment is far from benign in its effects on space systems. Automated and robotic systems, because of their complexity and autonomy, will in particular be threatened by environmental anomalies as a result of adverse interactions in space. The coupling of electrical transients into control systems, due, for example, to EMI resulting from plasma interactions and solar array arcing, may cause a variety of spurious commands that could be difficult to detect and correct in time to prevent damage during critical operations where human intervention might not be possible. As another example, spacecraft glow and space debris could introduce false imaging information into optical sensor systems causing guidance errors. This presentation will provide a brief overview of the more critical of these adverse environments (plasmas, neutral atmospheres, fields, and solid particulates) that could cause such interactions. Given the growing complexity and consequent sensitivity of automated and robotic space systems, an understanding of these environments and their interactions will be crucial to mitigating their effects.

ENVIRONMENTS

Introduction

In this review of the ambient space environment, 8 environments will be considered. The first, the neutral atmosphere, is primarily responsible for drag, glow, and oxygen erosion. The next 2 environments, the Earth's magnetic and electric fields, are responsible for magnetic torques and induced electric fields. The third electromagnetic environment to be discussed, the UV/EUV radiation environment, is not only responsible for the formation of the ionosphere but also for photoelectrons and long term changes in material properties. 4 plasma environments will be discussed: the Interplanetary Environment, the Plasmasphere/Ionosphere (responsible for ram/wake effects and solar array arcing), the Plasmasheet (the primary region for spacecraft charging) and it low altitude extension the Auroral Zone, and the Radiation Belts (the source environment for radiation dosage effects). Finally, the particulate environment, both man-made and meteoroid, will be briefly discussed. The intent is not to provide a detailed description of each of

these environments but rather to provide an overview of their chief characteristics as they apply to environmental interactions.

Neutral Atmosphere

By far the major environmental factor at shuttle altitudes is the earth's ambient neutral atmosphere. Whether it be through drag or the recently discovered interactions with atomic oxygen (glow and oxygen erosion), the effect of the neutral atmosphere (predominately the neutral atomic oxygen) on spacecraft dynamics and surfaces greatly exceeds any of the other effects that will be considered in this report. The 3 main sources of data at these low altitudes have been neutral mass spectrometers, accelerometers, and orbital drag calculations. Without going into detail, most models attempt to fit the observations with an algorithm that includes the exponential fall off of the neutral density, the effects of increasing solar activity (particularly in the ultraviolet), the local time, and geomagnetic activity. Of these, the large variations associated with increasing geomagnetic activity (and subsequent heating of the atmosphere) have eluded adequate modelling by this fitting process. Unfortunately, it is clear from many sources that these variations, particularly in density over the auroral zone, often dominate the neutral environment and that to date no adequate method of including these effects in the models has been devised (some recent very sophisticated theoretical computer models do hold promise, however). Here, some of the dominate features of the vertical and horizontal variations in density will be presented.

Consider first the vertical variations in the neutral atmosphere. In Fig. 1 (Carrigan and Shrivastava [1974]), the variations of the neutral atmosphere at orbital altitudes between about 100 km and 1000 km are plotted. This is, for example, the same region in which vehicle glow has been observed. For purposes of this review and for most practical applications, the neutral atmosphere can be considered according to Fig. 1 to consist mainly of atomic oxygen (note: atomic hydrogen can dominate occasionally above 500 km for low exospheric temperatures) with traces of molecular oxygen, molecular nitrogen, and atomic hydrogen over the altitude range of interest. Helium, nitric oxide, atomic nitrogen, and argon are also present below the one percent level (for general descriptions of the upper atmosphere, see Whitten and Poppoff [1971], Banks and Kockarts [1973] and references therein; widely used neutral atmosphere models are those of Jacchia [1972], Jacchia [1977], and Hedin [1987]). The thermal temperature of the constituents varies approximately exponentially from ~100 K at 100 km to 500-1500 K at 1000 km depending on solar cycle, latitude, and local time with excursions to 2000 K during high levels of geomagnetic activity. As spacecraft between 100 and 1000 km are moving at about 7.8 km/s, the resulting impact energy of the particles can reach values on the forward (or ram) surface of the spacecraft well in excess of 5 eV (varying from 4.6 eV for N to 10.25 eV for O2). These ram energies are sufficiently high to induce chemical reactions (including oxygen erosion). Further, the large ratio of the directed velocity to thermal velocity means that pronounced anisotropies exist in the flux to the vehicle. This has led to gross asymmetries in the glow phenomenon--surface glow appearing primarily on surfaces which face into the vehicle velocity vector.

2 types of models are often used to compute vertical profiles such as presented in Fig.1. These are the Jacchia family of models (i.e., Jacchia [1972], Jacchia [1972]) and the MSIS models (Hedin et al. [1977a]; Hedin et al. [1977b]; Hedin [1987]). These models are readily available in computer format and have been well developed over the last decade. For comparison, the more recent MSIS 1986 (Hedin [1987]) model and it companion, Jacchia 1977 (Jacchia [1977]), which are based on in-situ spacecraft measurements, deviate by about 20% from the older, drag based, Jacchia 1972 values on the average--a relatively small value given the much larger average uncertainties in the models themselves. These models, as illustrated in Figs. 2, 3, and 4 for the northern hemisphere and 400 km, can also be used to investigate horizontal variations in density, temperature, and composition. As can be seen, there is typically a two-fold increase in density from midnight to noon. Further, there is a pronounced shift by 2 hours of the peak in the density and temperature maxima away from local noon. This well know phenomena results from the rotation of the Earth and causes the peak in atmospheric heating to occur after local noon.

A typical shortcoming of the older, drag based models is that there are no clear density/temperature features associated with the auroral zone. This is directly due to the averaging techniques used in deriving models of this type which smooth out the density waves actually observed over the auroral zone--the more recent

MSIS 1986 and Jacchia 1977 do demonstrate auroral variations (these have been incorporated into a simple update of the Jacchia 1972 model by Slowey [1984]). An example of the variations at 120 km for the Jacchia substorm correction is presented in Fig. 5. Such correction factors, however, are only meaningful in an average sense--actual substorm variations can be an order of magnitude larger instanteously. Even so, the model results are useful in estimating the levels of atmospheric drag, shuttle "glow", and surface degradation.

Electromagnetic Environments

Magnetic and Electric Fields

Because of the rapid variations in the Earth's magnetosphere, the outer magnetic field beyond about 6 R_e is not precisely modelled. The approximate structure in the noon-midnight meridional plane is illustrated in Fig. 6 along with the major plasma regions that it delineates. In contrast, the geomagnetic field below 6 R_e and at Space Station/Shuttle altitudes is pretty accurately known. It can be crudely modelled, for example, in terms of a tilted (~11° from geographic north) magnetic dipole of magnitude 8×10^{25} G-cm^3. Numerous, very accurate models of this field exist such as the International Geomagnetic Reference Field for 1985 (IGRF [1986]). As a typical example of these models, consider the somewhat simpler POGO magnetic field model (Knecht [1972]; Cain and Langel [1968]) which is the basis of the International Reference Ionosphere (IRI) and various radiation models. As is characteristic of most models of the near-Earth magnetic field, this model is a straight forward expansion of fits to the Earth's magnetic field in terms of spherical harmonics. As shown in Fig. 7 for 400 Km, the field varies in this model from a minimum of .25 G near the equator to .5 G over the polar caps. 2 peaks exist in the magnitude of the magnetic field (over the north pole, these are at 270°E and 90°E) and reflect the true complexity of the magnetic field in the auroral/polar cap regions (note: if vector components are considered, the maximum at 270° east longitude is the true "dip" magnetic pole). Likewise, there are two minima near the equator--the largest of these, the so-called South Atlantic Anomaly--will be important in our discussions of the radiation belts. Finally, it should be noted that geomagnetic storm variations are superimposed on this main field. These are typically less that .01 G so that even during a severe geomagnetic storm, magnetic fluctuations are small compared to the average field--a marked contrast with the atmospheric and ionospheric environments!

Besides magnetic torques (which are very system dependent), the earth's magnetic field can induce an electric field in a moving body by the vxB effect:

$$E = 0.1\ (v \times B)\ V/m = .3\ V/m$$

at 400 Km

where:

v = spacecraft velocty = 7.6 km/s

B = .3 G

For the Shuttle, which is roughly 15 m x 24 m x 33m, potentials of 10 V could be induced by this effect. As systems grow to km or large dimensions, the induced fields will grow accordingly.

The induced electric field for a vehicle of 90° inclination are much more complicated than those for an equatorial orbit and, as would be anticipated, the largest electric fields are seen over the polar caps. Typical absolute values for a polar orbiting vehicle are presented in Fig. 8. In addition to these fields at polar latitudes, the ambient environment can also produce strong electric fields in the auroral regions. These fields can reach values of nearly 100 mV/m (Foster [1983])--a sizable fraction of the induced field. The fields are also comparable to the fields necessary to deflect charged particles in this environment as the particles have ambient energies of typically .1 eV (ram energies for the ions like oxygen can reach several eV, however) and thus must be taken into account when studying ionospheric fluxes.

UV/EUV (and X-Rays)

Solar ultraviolet (UV), extreme ultraviolet (EUV), and X-ray radiation are not only important to atmospheric and ionospheric dynamics but, through material surface changes and photoelectron emission, provide a major environmental factor for spacecraft at all altitudes. By UV/EUV radiation, we mean here the continuum and line spectrum between roughly 10 Å and 4000 Å. The energy in this spectral range is represented by a solar flux between 10^7 and 10^{10} photons/(cm^2 s) below 1000 Å. The flux rises

almost exponentially to 10^{16} photons/(cm^2 s) between 1000 Å and 10000 Å. The flux is not constant but varies in time due to a number of factors, one of which is the solar cycle variability. This radiation spectrum is also a complex variable of the atmospheric attenuation as a spacecraft moves in and out of the Earth's shadow (see Garrett and Forbes [1981]). An average spectrum is presented in Fig. 9 (Grard [1973]).

The shortest wavelengths, 10 Å-100 Å or less, are referred to as X-rays. This spectral range contributes to the ionization of the E-region. The spectral region from about 100 Å to 1000 Å, called EUV, is related to the photoionization processes of O_2, N_2, and O in the ionosphere and to thermospheric heating. UV radiation is the continuum and line spectrum between roughly 1000 Å and visible. This spectral region contributes to photo-dissociation, absorption, and scattering processes in the mesosphere, stratosphere, and troposphere. The spectral range from 10-1750 Å is absorbed in the lower thermosphere and effects the production of oxygen atoms and their vertical distribution above the mesopause. The Lyman-alpha line at 1216 Å plays a major role in the mesosphere through the disassociation of O_2, H_2O, and CO_2 and the ionization of nitric oxide. The spectral region between 1750 to 2400 Å leads to the dissociation of O_2 and to ozone production in the mesosphere and stratosphere. Between 2400 A and 3300 A, the solar irradiance is responsible for the disassociation of ozone and other trace gases that play a role in the stratospheric budget.

Plasma Environments

Introduction

There are 6 plasma regions with distinct characteristics that normally need to be considered in defining the Earth's magnetosphere or plasma environment. These are illustrated in Fig. 6. First there is the Solar Wind, a flowing magnetized plasma emitted by the Sun, that is responsible for shaping the Earth's magnetosphere. During severe geomagnetic activity, it has been observed within geosynchronous orbit but, typically is observed outside of about 10 R_e on the dayside. The region marking the transition between the Solar Wind and the Magnetosphere is called the Magnetosheath (we will not treat this region here but consider it as a subset of the Solar Wind for interactions purposes). Within the Magnetosphere, at low latitudes and altitudes, is the ionized extension of the Earth's atmosphere--the Ionosphere or, above 1000 km, the Plasmasphere. This cold plasma environment is typical of the Space Station environment. At higher latitudes and radial distance lies the Plasmasheet and its lower boundary, the Auroral Zone. This is the hot plasma primarily responsible for spacecraft charging. At still higher latitudes lies the Polar Cap environment. Normally this environment closely resembles the Ionosphere/Plasmasphere environment--only during solar flare or proton events, when energetic particles can gain direct access to the polar caps along magnetic field lines connected to the Solar Wind is this environment substantially different (here it will be considered a subset of the Ionospheric environment). Finally there is the very energetic Van Allen radiation belts that overlay the Plasmasphere and Plasmasheet. As these particles show significantly different time variations than the other two plasma environments, they will be treated as a separate population here.

Interplanetary Environment

Sun

The source of virtually all space disturbances is the Sun. Through a poorly understood process deep within the Sun, strong magnetic fields, thousands of times stronger than the Earth's magnetic field, are generated and brought to the solar surface. Material motions on or near the solar surface twist and shear these fields, ultimately producing instabilities in them. The fields in turn are responsible for the fundamental processes which affect space operations. Although the visible surface of the Sun is at a temperature of abut 6000 K, the magnetic fields serve as a conduit channelling energy into the outer solar atmosphere. This outer atmosphere, the solar corona visible during a solar eclipse or directly observable from space, is therby heated to temperatures over 10^6 K. At this high temperature the solar corona "boils" off into space at velocities of about 500 km/s, carrying some of the imbedded magnetic field with it. This "solar wind" is highly time variant with a complex magnetic structure. When the solar wind interacts with the Earth's own magnetic field some three to five days after it left the Sun, space environmental problems are born.

A far more energetic process, and potentially damaging situation, occurs when very strong magnetic fields in the solar corona reach a critical

instability. On times scales of seconds the strong fields are unstable enough to "snap" thereby adjusting and relaxing to remove the instability. A considerable amount of energy, up to .1% of the total solar enery output or about 10^{32} ergs, is released during this "flare". A solar flare typically lasts from a few minutes to a few hours and heats the surrounding corona to temperatures in excess of $2x10^7$ K. Associated with the heating, large fluxes of atomic particles, electrons, neutrons, and protons, are accelerated and expelled from the Sun. There are also substantial radio bursts and X-ray emissions. The X-ray emissions and radio bursts arrive at the Earth in 8 minutes and are the first evidence at the Earth that a solar flare has occured. About an hour later the highest energy particles, mostly neutrons, protons, and heavier atomic nuclei, arrive. However it is the flare blast wave or solar wind shock, consisting of charged particles (electrons), and carrying with it portions of the tangled coronal magnetic field, arriving several days later at the Earth which represents the most serious hazard for space systems in Earth orbit (as opposed to systems in interplanetary space where single event upsets due to the solar flare protons are the most serious problem) as this shock can initiate a major geomagnetic storm and can lead to pronounced enhancements of the radiation belts.

The Sun has a roughly periodic activity cycle. Due to poorly understood phenomena deep within the solar interior, solar magnetic field penetration and eruption through the solar surface has an eleven year cycle. Although the solar wind is fairly constant during this cycle, solar flare related phenomena occur far more frequently during solar maximum (in terms of sunspot number). The current solar cycle, which is particularly severe, is expected to peak in early 1990. There is some evidence that changes in overall solar energy output of up to 1% are also associated with the solar cycle. These variations are enough to affect global terrestrial weather patterns. The solar cycles themselves, while fairly predictable in time, have peak activity levels which may vary by factors of four from one maximum to the next (Fig. 10). Moreover, historical records have indicated relatively long periods, for example, during most of the 18th century, when there were no discernable solar cycles. It is, however, the effects of solar flares and geomagnetic storms that most impact the Space Station and Shuttle environments. Although solar flares are far less frequent during solar minimum, flare related effects can be greater in the terrestrial magnetosphere because the interplanetary fields, through which the flare material must travel, are less complicated and solar flare particles can more easily gain access to the earth's polar caps. Geomagnetic activity at the Earth also shows this variation. Although the number of geomagnetic storms goes down during minimums in solar activity, the level of a given storm, even during the lowest levels of solar activity, can be among the highest ever seen.

Solar Wind

The dominate environment in the Solar System is that of the solar atmosphere or heliosphere--the Solar Wind--and, while it does not directly contribute to Shuttle and Space Station interactions, it the primary energy source of geomagnetic activity which does. As discussed, the Solar Wind is the low density plasma (predominately hydrogen ions with some helium) that is being continuously emitted from the solar corona at supersonic speeds. The plasma is characterized by a residual magnetic field (typically a few 10's of nT; 1 nT= 1 nano Tesla or 1 gamma) and variable velocity and density. The Solar Wind velocity vector is observed to be dominantly in a radial direction in the ecliptic plane with a magnitude of 200 to 500 km/s. Since the Sun rotates with a period of 27 days, the Solar Wind takes on a spiral structure with the spirals marked by regions of similar magnetic polarity (Fig. 11). At present, based on in-situ measurements from the Pioneer spacecraft, we know that this environment extends out to and beyond the orbits of Pluto and Neptune where at some point it terminates in the interstellar medium. Typical plasma values for the Solar Wind are for distances corresponding to near Mercury 50 cm^{-3} and 50 eV, 2 cm^{-3} and 10 eV for the ions and 50 eV for the electrons at the Earth, and .2 cm^{-3} and 1 eV for the ions and 10 eV at Jupiter. The Solar Wind represented by these values can be a significant source of spacecraft charging in the interplanetary medium. Finally, it should be noted that it is now accepted that there is a direct relationship betweent the direction of the Solar Wind magnetic field and geomagnetic activity--when the Solar Wind field points southward, the likelihood of geomagnetic activity is greatly enhanced. The reason for this connection is not completely understood but is generally believed to be due to increased coupling and energy transfer between the Solar Wind magnetic field and the Earth's geomagnetic field.

Plasmasphere/Ionosphere

Illustrated in Figs. 12 and 13 (Carrigan and Skrivanek [1974]) are various vertical profiles of the ionosphere. On the sunlit hemisphere of the Earth, X-rays, EUV, and UV radiation penetrate the neutral atmosphere, ionizing and exciting the molecules present. As the radiation penetrates, there is a balance between increasing neutral density and increasing absorption that leads to the formation of ionized layers (principally the F layer between 150 and 1000 km, the E layer between 100 and 150 km, and the D layer between 60 and 100 km) that gives rise to the mean structure called the ionosphere (for general descriptions of the upper ionosphere, see Whitten and Poppoff [1971], Banks and Kockarts [1973] and references therein; a widely used ionospheric model is the International Reference Ionosphere--IRI-86; Rawer and Bradley [1987]). These layers are the combined result of the absorption/increasing-density process and complex chemical reactions within the atmosphere and ionosphere. As illustrated by the horizontal profiles in Fig. 14 from the IRI model, the local time peak in the ionospheric density parallels that of the neutral density bulge--occurring approximately 2 hours after local noon. The ionospheric composition likewise follows that of the neutral atmosphere, varying roughly from NO^+/O_2^+-dominated in the D region, to O^+-dominated in the E region, to H^+-dominated in the F region (chemical reactions complicate the picture). Densities reach 10^6 cm^{-3} at the peak in the F region at about 300 km on the sunlit side. At night, the peak ion density can fall below 10^5 cm^{-3} and the composition change from O^+ to H^+ above 500 km. Temperatures follow roughly that of the neutral atmosphere--increasing exponentially from a few 100 K at 50-60 km to 2000-3000 K above 500 km (i.e., a few tenths of an eV). The electron temperature tends to be a factor of two greater than that of the neutrals, with the ion temperature falling in between.

In order to systematically evaluate the effects of the ionosphere on automated and robotic systems, fairly detailed models of the Space Station environment are required. At present relatively few ionospheric models are available and most of these only predict electron densities--the most readily measureable quantity by ground means and the most important to radio propagation. Unfortunately, the ionospheric composition is particularly critical to adequate modelling of the Space Station ram/wake. The principle ionospheric model currently in use is the International Reference Ionosphere (Rawer and Bradley [1987]). This computer model, based primarily on ground based observations of the total electron content, is the only readily available computer model that gives both the electron and ion composition and temperature as a function of longitude, latitude, altitude (65 to 1000 km), solar activity (by means of the sunspot number, R), and time (year and local). Although the model is limited (it is confined to R values of 100 or less whereas R values of 200 may occur during solar maximum), it is the "best" available comprehensive model of the ionosphere. As an example of the output, the model predicts that, unlike the neutral temperature, the electron temperature increases by a factor of 2 in going from the equator to the pole (Fig. 14). The IRI computer simulation of the ionosphere shows a complex local time variation with the peak in electron density on the day side and/or in the auroral zones. These variations in turn lead to pronounced variations in the ram/wake structure of the Space Station that can cause interactions with systems operating in and near the Space Station or a similar large structure.

A major shortcoming of current ionospheric (and atmospheric models also) is their inability to properly model high latitude geomagnetic effects. As will be discussed, above about 60° geomagnetic latitude, the Earth is subjected to intense fluxes of high energy electrons and ions from the magnetosphere of the Earth and by direct entry of solar flare particles. These particles (typical energies between 100 eV and 10 KeV) generate considerable ionization that can easily excede the UV/EUV levels at these latitudes below 1000 km. Unlike the fairly constant UV/EUV fluxes, these corpuscular precipitation events (as they are some times called) can vary greatly in time--often occurring in less than a 1/2 hour. They are intimately associated with the auroral displays that are seen in the polar regions and indeed they are the cause. The ionization can increase orders of magnitude in a similar short time period adding great complexity to the polar ionosphere. Currently, only a few very complex "Thermosphere General Circulation Models" (TGCMs) can adequately model these effects. As yet it very hard to apply the results of these models to practical examples as the results are too complex to readily interpret.

Aurora/Plasma Sheet

The flow of the magnetized Solar Wind plasma past the Earth's magnetic field creates a long (perhaps 1000's of R_e in length) magnetic cavity in the antisunward direction (see Fig. 6). Inside this cavity, extending roughly from about geosynchronous orbit to the magnetopause in the sunward direction and from geosychronous along the length of the magnetic tail in the antisunward direction, is a thick "sheet" of warm plasma. This plasma sheet, as it is called, is believed to extend earthward from geosynchronous orbit along the magnetic field lines that map into the auroral zones in the northern and southern hemispheres (roughly between 55° and 75° depending on geomagnetic activity). The "temperature" of the plasma ranges between 100's of eV to 10's of keV for the electrons and 1000's of eV to 10's of keV for the ions (primarily hydrogen ions with variable concentrations of oxygen ions). The density is quite variable, changing from less than 1 to 10's of particles per cubic centimeter in minutes or less. The spatial location of the inner plasma sheet boundary is roughly between 5 and 8 R_e--varying greatly with geomagnetic activity and spatial location. The plasma sheet is marked by rapid temporal fluctuations called geomagnetic substorms (or, more precisely, "injection events") which compress it inward in the equatorial plane and increase its density and to a lesser extent its average energy. These events have been found to correlate with sharp increases in surface charging on geosynchronous spacecraft and represent the major source of surface charging within the Solar System. The events are believed to manifest themselves as aurora at lower altitudes--the plasma sheet serves as a giant electron beam source which paints the Earth's upper atmosphere which then acts like an oscilliscope screen, the aurora being the "trace". It is indeed these high latitude aurora which pose the greatest threat to low altitude, polar orbiting systems. A typical spectrum is presented in Fig. 15 (Gussenhoven et al. [1985]).

The most dramatic changes in the Earth's environment at Space Station altitudes are brought about by geomagnetic substorms. These changes are reflected in visible auroral displays and in intense particle and field variations in the auroral regions down to 100 km. The precipitating particle patterns (chiefly electrons as the ions are typically scattered high in the atmosphere by various processes) can be roughly divided into a broad, diffuse background and discrete features. Simple auroral flux models (Hardy et al. [1985] and Fuller-Rowell and Evans [1987]) of the diffuse populations are available. They predict a .1-1 keV particle population flowing down along magnetic field lines into the upper atmosphere. The pattern can be approximated by a simple sinusoidal variation in geomagnetic local time, a gaussian in latitude (the peak occurring near 65°-75°), and a roughly linear increase in the geomagnetic Kp index. Superimposed on this are the so-called discrete aurora characterized by latitudionally narrow features (some believed to be smaller than a km) and great longitudional extent (10's of degrees). The particles, again primarily electrons, have energies in the 1 keV to 10 keV range. Even though the flux of these particles can reach 10's of nA, the ambient flux due to the cold plasma running into the Shuttle approaches mA. Only above about 700 km, and then rarely, can the auroral fluxes excede the ionospheric density. It is, however, in the wake of the Shuttle or other large structure as it passes through the aurora that problems can occur. In the wake, as the ambient ionospheric ions can not penetrate effectively (the cold ions, primarily O^+, have thermal velocities of only 2-3 km/s whereas the orbital velocity at Space Station altitudes is on the order of 7 km/s) and keep the cold electrons out, only the high energy electrons associated with the aurora can penetrate. They thus determine the potential of electrically isolated surfaces in the wake.

Radiation

Introduction

The high energy radiation environment will be assumed here to consist of electrons with energies greater than 100 KeV and protons with energies greater than 1 MeV but with energies below 1 BeV. Above about 500 MeV energy, the particles are typically considered solar flare particles or Cosmic Rays--these will be considered separately. Heavy ions associated with solar flares will also be briefly mentioned. It should also be noted that there is increasing evidence that heavy ions, particularly O^+, are common if not dominant components of the Earth's radiation belts, but the measurements are still being evaluated. Currently, this radiation regime is defined in terms of two sets of models--the National Space Science Data Center (NSSDC) AE and AP models (these are further divided into versions at Solar Maximum--AE8MAX and APMAX--and Solar Minimum--AE8MIN and AP8MIN). Therefore, rather than complicate the definition of this environment with a discussion of the many variations typical of the regime, we will, following a description of the form of the AE/AP models, define the radiation environment in terms of them.

Trapped Particles

There are potentially many different ways to model the Earth's radiation environment. Fortunately, the use of adiabatic invariants, the introduction of the McIlwain's B-L coordinates, and the predominance of the AE/AP series of radiation models have produced a uniform set of practical models. For the purpose of studying radiation effects on long duration space missions, these AE/AP models produced by the NSSDC have proven to be very useful. Here the major characteristics of the radiation environment will be summarized in terms of these AE/AP type of models. It should be remembered, however, that there are other means of representing the environment that may be more appropriate for specific uses. The AE/AP models assume that the flux of particles can be given as a function of B and L coordinates integrated over some energy interval. Units are typically "particles per square centimeter per second" with the integrated energy channel being from the stated energy (typically 40 keV to 4 MeV for electrons in the AE electron models and 1 to 400 MeV for the AP proton models). A valuable attribute of these models is that, in principle, it is possible to construct this type of simple model by one measurement of the fluxes at all pitch angles as a satellite moves away from the Earth in the magnetic equatorial plane. As modelling of the Earth's radiation environment is a complex process at best, the AE/AP do have some limitations--for example, the effects of inadequate data coverage and the lack of recent data. The NSSDC has recently issued the AE8/AP8 versions of the model which correct some of these inadequacies.

The major difficulty in developing a model of the Earth's radiation belts is that both space and time must be factored into the model. Although the use of the adiabatic invariants and B-L coordinates are very useful in simplifying this task, in reality asymmetries in the Earth's magnetic and electric fields and their time variations introduce significant complications into the modelling process. In particular, due to such effects as "shell-splitting" (particles of the same energy but different pitch angles tend to follow slightly different drift paths around the Earth so that particles observed together at the equator on one side of the Earth are separated in radial distance on the other side), distortions in the Earth's magnetosphere, and similar effects, force the inclusion of time and local-time (or, less precisely, longitudional) variables. The AE/AP model fluxes, for example, can be parametrically represented by:

$$J(>E,B,L,\theta,T) = N(>E,L)\Phi(>E,L,\theta)G(B,L)$$

where J is the integral omnidirectional flux, >E means for all energies above E, θ is the local time, and T is the epoch (or date). Data from many different satellites are averaged in discrete B and L bins to determine the B-L variation G, in energy, L, and local time to determine the local time variation Φ (note: B variations were ignored because there was often too little data to allow simultaneous binning in terms of B also), and in energy and L bins to determine the energy variations N.

Fig. 16 for 1 MeV electrons and 10 MeV protons from the AE8 and AP8 models respectively illustrates the basic structure of the radiation belts. In particular, the electron contours show a dual peak (the protons have a similar structure but the division between the peaks is less obvious). Typically, therefore, the radiation belts are divided into "inner-zone" and "outer-zone" populations. This division also roughly corresponds, for the electrons, to an inner belt which is weakly affected by geomagnetic storms and an outer belt which is greatly affected by storms. The L-shell region up to L~2.5 is termed the inner-zone while the region beyond L~3 is considered to be the outer-zone with a "slot" region of reduced density in between. The inner-zone electrons peak around L = 1.45 to 1.5 (typical values at 1.45 are: $>10^8$ for $E \geq 0.1$ MeV; $>10^6$ for $E \geq 1$ MeV; $>10^5$ for $E \geq 2$ MeV; integral, omnidirectional flux in units of $cm^{-2}\ s^{-1}$). Little variation with geomagnetic activity is seen below L~1.6. The inner-zone protons are very stable, varying inversely with atmospheric density (the fluxes are lower at solar maximum when the atmospheric density is highest). The proton flux peaks near L = 1.45 (typical values at 1.45 are: $>10^4$ for $E \geq 100$ MeV and $>10^3$ at 300 MeV). In the outer zone, the peak L shell varies with energy for the electrons (typical values are: $>10^8$ for $E \geq 0.1$ MeV, L = 6; $>10^7$ for $E \geq 1$ MeV, L = 5; $>10^5$ for $E \geq 4$ MeV, L = 4; units are as above). Flux increases can be as great as 10^3 in less than a day during a geomagnetic storm. The outer-zone protons do not show as strong a division into an outer belt as the electrons nor as much variation with geomagnetic activity. Protons with $E \geq 1$ MeV peak at about L = 3 while protons with $E \geq 10$ MeV peak at about L = 2.5 (typical intensities are: $>10^8$ for $E \geq 0.1$ MeV; $>10^7$ for $E \geq 1$ MeV; $>10^5$ for $E \geq 10$ MeV; $<10^2$ for $E \geq 100$ MeV; units as above). (Note: all numbers are adapted from Vampola [1989].)

Galactic Cosmic Rays and Solar Flares

The AE/AP models only describe the trapped electron and proton environments. Recently, with increasing concern over single event upsets, there has been a need for models of the high energy, high charge number ions above hydrogen in mass that are primarily associated with solar flares. These particles, generally associated with solar flare proton events and Galactic Cosmic Rays, can cause bit flips in sensitive electronic components without damaging the components. These logic changes can seriously affect a satellite's control systems if permitted to go uncorrected. As yet, models capable of accurately predicting the occurrence frequency of solar flares are not reliable. Recent models by Feynman et al. [1989] of solar flare proton events, however, hold the key to future progress in this area. Here, instead, the physical characteristics of the events will be presented.

Consider first Galactic Cosmic Rays. Galactic Cosmic Rays (GCR) are primarily interplanetary protons and ionized heavy nuclei with energies from ~1 MeV/nucleon to higher than $\sim 10^{10}$ GeV/nucleon. Electrons are also a constituent of GCR, but their measured intensities at energies above ~100 MeV are at least 1 order of magnitude smaller than that of the protons and are usually ignored. Experimental studies indicate that cosmic ray fluxes are isotropic over the entire energy range, suggesting that they are galactic and/or extragalactic in origin. Figs. 17 and 18 display the observed cosmic ray energy spectrum and abundance distribution for the chemical elements in the energy range of ~100 MeV/nucleon to ~1 GeV/nucleon from hydrogen to the iron group (Meyer et al. [1974]). For comparison, solar system abundances are also shown in the figure. Note that the two abundance distributions are strikingly similar.

In addition to Galactic Cosmic Rays, hydrogen and heavy nuclei in the ~1 MeV/nucleon to ~10 GeV/nucleon energy range are ejected during a solar flare. Their intensities are generally a few to several orders of magnitude larger than those of Galactic Cosmic Rays at these lower energies, depending on the size of solar flare. Detailed studies of solar flares are limited by a current lack of sufficient data, particularly that of the relative elemental abundances, and their unpredictable occurrence frequency. Solar flare particles (Galactic Cosmic Rays also, but their fluences are typically too low to be of concern), because of their high penetrating abilities and ionizing powers, are known to induce single event upsets (SEU) and other malfunctions in digital microelectronics devices even at Space Station altitudes. A series of reports by J. Adams and his collaborators at Naval Research Laboratory (NRL) provide the relevant formulations and information on this important subject (Adams et al. [1981]; Adams et al. [1983]; Tsao et al. [1984]; Adams et al. [1986]). In these reports, in addition to the Galactic Cosmic Ray model, a worst-case solar flare model is developed based on the combined features of the two largest solar flares ever observed (one in 1956 and one in 1972). A comparison between the galactic cosmic ray flux and the worst-case solar flare proton event in 1972 is presented in Fig. 19. As shown, the worst-case solar flare proton flux is ~5 orders of magnitude larger than the galactic cosmic ray flux, but becomes "softer" above ~10 GeV.

In evaluating the effects of Galactic Cosmic Rays and solar flare particles on spacecraft, one final point needs to be considered. That is, for the low Earth orbit environment, the geomagnetic field provides shielding against incident Galactic Cosmic Rays and solar flare particles as it can effectively deflect (through the Lorentz force) the lower energy cosmic ray and solar flare particles. Because of the approximate dipole nature of the geomagnetic field, vertical particle velocities in the polar regions are essentially parallel to the magnetic field resulting in almost vanishing Lorentz force so that the particles can gain direct access. At low inclinations, only particles with sufficiently high energy, or "rigidity", can penetrate through the magnetic shielding.

Particulates

Introduction

Although the primary effect of the ambient, macroscopic particulate environment represented by space debris and interplanetary meteoroids is mechanical (i.e., impact craters), it can also indirectly change the EM characteristics of spacecraft systems in several ways. For instance, penetration of insulation can result in "pinholes" that expose the underlying conductors to the plasma environment. The subsequent current collection and related effects may seriously alter the local surface fields. The ejecta cloud produced by an impact can be partially ionized, causing charging and/or an electromagnetic pulse. Also,

cumulative erosion effects will eventually result in failure of exposed insulation, solar array surfaces, and wiring. The sources of these particulates in Earth orbit are both extraterrestrial--comets--and terrestrial--waste products and the remains of previous satellites. Given that the manmade debris is already pervasive in low Earth orbit, it will be considered as part of the "natural", or pre-existing, environment. Each of these particulate environments will be separately considered.

Interplanetary Meteoroid Environment

The close of the decade of the 60's saw the completion of three definitive NASA design criteria documents on the meteoroid environment and its effects on spacecraft systems. As of this date, the models presented in those documents still represent the official NASA meteoroid environment despite more recent data on the in-situ environment. The three documents are:

1.) "NASA Space Vehicle Design Criteria (Environment); Meteoroid Environment Model (Near Earth to Lunar Surface)", NASA SP-8013 [1969].

2.) "NASA Space Vehicle Design Criteria (Environment); Meteoroid environment Model (Interplanetary and Planetary)", NASA SP-8038 [1970].

3.) "NASA Space Vehicle Design Criteria (Structures); Meteoroid Damage Assessment", NASA SP-8042 [1970].

The first document defines the meteoroid environment between the Earth's surface and the Moon and is the primary model used for the interplanetary meteoroid flux. It provides working definitions of the three principle quantities needed to define the meteoroid environment: their mass versus number density, their velocity distribution, and their composition. Included in the document are listings of interplanetary meteor streams and the Earth-based meteor observations on which the models are based. The second document presents an extrapolation of the Earth-based observations to interplanetary space for meteoroids of both cometary and asteroidal origin. The final document outlines ground studies of high velocity impacts, methods of modeling meteoroid penetration, and techniques for limiting meteoroid damage. Of the three documents, the first is the primary sources of the material to be presented here and will be discussed in more detail.

Meteoroids as defined by the NASA documents are solid particles orbiting in space that are either of cometary or asteroidal origin. The spatial volume of interest ranges from 0.1 to 30.0 astronomical units (AU). The mass range is from 10^{-12} to 10^{22} g. Knowledge of these particles is based primarily on Earth-based observations of meteors, comets, asteroids, the zodiacal light, and in-situ rocket and spacecraft measurements. In all cases, the flux versus mass of the particles, the basic quantity required to model the meteoroid environment, is not directly measured but must be inferred (e.g., from light intensity, crater distributions, etc.). The ground-based measurements consist principally of photographic and radar observations. These yield fluxes for masses from 10^{-3} g or larger and 10^{-6} to 10^{-2} g respectively. Observations of the zodiacal light and direct in-situ measurements cover a much smaller mass--typically 10^{-13} to 10^{-6} g. At the other extreme, telescopic observations of asteroids and planetary and lunar crater counts are used to determine the distribution from 50 km and up. As should be obvious, there are large data gaps in the assumed distribution. Of most concern to the Space Station is the range from about 10^{-3} to 10 g as these particles pose the major threat of catastrophic failure to crew modules. Of more concern for pinholes, the major problem for EMC, is the mass range from 10^{-3} down to 10^{-9} g as these particles will have sufficient flux to erode surfaces and sufficient energy to penetrate protective coatings. Meteorites fall in this range but due to the infrequency of actual observed impacts and the difficulty of relating the final mass to the original mass, little data are available. Sufficient information does exist to justify dividing the observations into two groups: those of cometary origin (average density around 0.5 g-cm^{-3}) and those of asteroidal origin (average density around 3.5 g-cm^{-3}), however, it is the cometary meteoroids which predominate near Earth.

Cometary Meteoroids

Cometary meteoroids in the mass range of interest (<10 g) are believed to be the solid remains of large water-ice comets that have long since evaporated or broken up due to collisions. The remaining silicate or chondritic material is of very low density (0.16 to 4 g-cm^{-3}) with an assumed value of 0.5 g-cm^{-3} for the NASA model. The primary flux of meteoroids inside 1.5 AU is made up of these cometary meteoroids as the denser asteroidal meteoroids are concentrated in the asteroid belts and peak at 2.5 AU. The

cometary integral mass distribution, derived from the observations discussed previously and assumed in the NASA Cometary Model, is plotted in Fig. 20. The velocity distribution relative to a massless Earth is presented in Fig. 21 for several different models (Morgan et al. [1988]) with the equivalent NASA model results plotted as a solid line ("massless Earth" means that the data have been corrected for the gravitational pull of the Earth).

NASA 8013 describes the total cometary meteor flux at Earth by the following:

$$\log_{10} F_c = -14.37 - 1.213 \log_{10} m$$

for $10^{-6} \leq m \leq 1$

$$\log_{10} F_c = -14.339 - 1.584 \log_{10}^2 m - .063 (\log_{10}^2 m)$$

for $10^{-12} \leq m \leq 10^{-6}$

where F_c is the number of cometary meteoroid impacts of mass m grams or larger per square meter per second. The gravitationally focused, unshielded flux, F_c, must be multiplied by a defocusing factor for Earth, G_E, as well as a shielding factor, ϵ.

The defocusing factor which corrects for the gravitational enhancement at a given distance from the Earth's center may be expressed as:

$$G_E = 0.568 + 0.432 (R_e/r)$$

The correction due to the physical presence of the Earth itself, which shields the spacecraft (randomly oriented) is expressed as:

$$\epsilon = 0.5 + 0.5 (1 - (R_e/r))^5$$

Multiplying by eta has the effect of subtracting out the flux within the solid angle subtended by the shielding body.

where:

- R_e = the Earth's radius
- r = the distance of the spacecraft from the center of the Earth

Space Debris

Increasing spaceflight operations in the Earth's vicinity have led to the creation of an artificial shell of debris around the Earth. This shell of debris poses an impact threat greater than the natural meteoroid environment within 2000 km of the Earth's surface. The Earth debris impact threat is credible and requires careful consideration. It is the purpose of this section to briefly review the basis of the existing debris model for spacecraft in low Earth orbit and to illustrate how debris impact calculations are carried out.

The main sources of orbital debris are orbiting spacecraft, fragments from exploded boosters or spacecraft, metal oxides and particulates from solid rocket motors, and ejected items from previous missions. These in turn collide with each other creating further debris--the Shuttle Challenger had one of its windows pitted by a debris particle, clearly illustrating the reality of such collisions. There are currently several sets of observational data for this growing threat. First, there are ground based optical and radar observations that form the bulk of the information. These are primarily from the US Space Command orbital element sets for objects of 10 cm diameter and larger, from optical measurements by MIT for objects 2 cm in diameter and larger, and from debris particle albedo measurements using an IR telescope at ATMOS/MOTIF, US Space Command radars, and NASA and Space Command telescopes. Second, for particles between 10^{-6} and 10^{-3} cm in diameter, in-situ measurements are available from sample surfaces retrieved from the Solar Maximum Mission (Laurance and Brownlee [1986]) at 500 km altitude. The NASA model and these observations are compared as a function of height in Fig. 22. Data from IRAS and other in-situ experiments are expected to further expand the debris data base in the near future.

Kessler (Kessler et al. [1989]) estimates the cumulative flux of debris on orbiting spacecraft to be given by:

$$F(d,h,i,t,S) = k \Phi(h,S) \Psi(i) (F_1(d) g_1(t) + F_2(d) g_2(d))$$

where:

- F = Flux in impacts per square meter of surface area per year
- k = 1 for a randomly tumbling surface; must be calculated for a directional surface
- d = Debris diameter (cm)
- t = Time (years)
- h = Altitude (km); h < 2000 km
- S = 13-month smoothed 10.7 cm wavelength solar flux (10^{-4} Jy); retarded by 1 year from t
- i = Inclination (degrees)

and

$$\Phi(h,S) = \Phi_1(h,S) / (\Phi_1(h,S) + 1)$$

$\Phi_1(h,S) = 10^{(h/200 - S/100 - 1.5)}$

$F_1(d) = 1.05 \times 10^{-5}\, d^{-2.5}$

$F_2(d) = 7 \times 10^{10}\, (d + 700)^{-6}$

$g_1(t) = (1 + 2\,p)^{(t-1985)}$

$g_2(t) = (1 + p)^{(t-1985)}$

p = Annual growth rate of mass in orbit = 0.05

Fig. 20 (Kessler et al. [1989]) compares the debris flux predicted by this model and the NASA SP-8013 [1969] meteoroid flux at an altitude of 500 km.

CONCLUSION

This paper has reviewed the key ambient environments that could potentially cause operational anomalies or damage to automated and robotic systems. These environments cover large portions of the electromagnetic and kinetic energy spectrum. As a result, these environments are difficult to protect against and will require mitigation techniques. In many situations, these protection methods have yet to be determined. Ultimately, the information presented here will be critical to the development of these methods. Hopefully, this review will prove to be a useful starting point in this process.

"The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration."

REFERENCES

Adams, J.H., Jr., R. Silberberg, and C.H. Tsao, "Cosmic Ray Effects on Microelectronics, Part I: The Near-Earth Particle Environment", NRL Memorandum Report 4506, Naval Research Laboratory, Washington, DC 20375, 1981.

Adams, J.H., Jr., J.R. Letaw, and D.F. Smart, "Cosmic Ray Effects on Microelectronics, Part II: The Geomagnetic Cutoff", NRL Memorandum Report 5099, Naval Research Laboratory, Washington, DC 20375, 1983.

Adams, J.H., Jr., "Cosmic Ray Effects on Microelectronics, Part IV", NRL Memorandum Report 5901, Naval Research Laboratory, Washington, DC 20375, 1986.

Banks, P.M., and G. Kockarts, "Aeronomy, Parts A and B", Academic Press, New York, 1973.

Cain, J.C., and R.A. Langel, "The Geomagnetic Survey by the Polar Orbiting Geophysical Observatories Ogo-2 and Ogo-4, 1965-1967", GSFC REP X-613-68-502, Greenbelt, MD, 1968.

Carrigan, A.L., and R.A. Skrivanek, "The Aerospace Environment", AFCRL Chart, 1974.

Daly, E.J., "The Evaluation of Space Radiation Environments for ESA Projects", ESA Journal, 12, 229-247, 1988.

Feynman, J., T. Armstrong, L. Dao-Gibner, and S. Silverman, "A New Interplanetary Proton Fluence Model", to appear in J. SPACECRAFT, 1989.

Foster, J.C., "An Empirical Electric Field Model Derived from Chatanika Radar Data", J. GEOPHYS. RES., 88, 981-987, 1983.

Fuller-Rowell, T.J., and D.S. Evans, "Height-Integrated Pedersen and Hall Conductivity Patterns Inferred From the TIROS-NOAA Satellite Data", J. GEOPHYS. RES., 92, 7606-7618, 1987.

Garrett, H.B., and Forbes, J.M., "A Model of Solar Flux Attenuation During Eclipse Passage and Its Effects on Photoelectron Emission from Satellite Surfaces", Planet. Space Sci., 29, pp. 601-607, 1981.

Grard, R.J.L., "Properties of the Satellite Photoelectron Sheath Derived from Photoemission Laboratory Measurements", J. Geophys. Res., 78, pp. 2885-2906, 1973b.

Gussenhoven, M.S., D.A. Hardy, F. Rich, W.J. Burke, and H.-C. Yeh, "High Level Spacecraft Charging in the Low Altitude Polar Environment", J. GEOPHYS. RES., 90, 11009, 1985.

Hardy, D.A., and M.S. Gussenhoven, "A Statistical Model of Auroral Electron Precipitation", J. GEOPHYS. RES. 90, 4229-4248, 1985.

Hedin, A.E., Jr., J.E. Salah, J.V. Evans, C.A. Reber, G.P. Newton, N.W. Spencer, D.C. Kayser, D. Alcayde, P. Bauer, L. Cogger, and J.P. McClure, "A Global Thermospheric Model based on Mass Spectrometer and Incoherent Scatter Data, MSIS 1, N2 Density and Temperature", J. GEOPHYS. RES., 82, 2139-2147, 1977a.

Hedin, A.E., Jr., C.A. Reber, G.P. Newton, N.W. Spencer, H.C. Brinton, H.G. Mayr, and W.E. Potter, "A Global Thermospheric Model based on Mass Spectrometer and Incoherent Scatter Data, MSIS 2, Composition", J. GEOPHYS. RES., 82, 2148-2156, 1977b.

Hedin, A.E., "MSIS-86 Thermospheric Model", J. GEOPHYS. RES., 92, pp. 4649-4662, 1987.

IGRF, "International Geomagnetic Reference Field Revision 1985", EOS, 523-524, 17 June, 1986.

Jacchia, L.G., "Atmospheric Models in the Region from 110 to 2000 km", in CIRA 1972, Akademie-Verlag, Berlin, pp. 225-328, 1972.

Jacchia, L.G., "Thermospheric Temperature, Density, and Composition: New Models", Spec. Rep., 375, Smithson. Astrophys. Observ., Cambridge, Mass., 1977.

Kessler, D.J., R.C. Reynolds, and P.D. Anz-Meador, "Orbital Debris Environment for Spacecraft Designed to Operate in Low Earth Orbit", NASA TM-100-471, 1989.

Knecht, D.J., "The Geomagnetic Field", Air Force Surveys in Geophysics, No. 246, AFCRL-72-0570, 26 Sept., 1972.

Laurance, M.R., and Brownlee, D.E., "The Flux of Meteoroids and Orbital Space Debris Striking Satellites in Low Earth Orbit", NATURE, 323, pp. 136-138, 1986.

Meyer, P., R. Ramaty, and R. Webber, "Cosmic Rays - Astronomy with Energetic Particles", PHYSICS TODAY, Oct., 1974.

Morgan, T.H., Zook, H.A., and Potter, A.E., "Impact Driven Supply of Sodium and Potassium to the Atmosphere of Mercury", to appear in ICARUS, 1988.

NASA SP-8013, "NASA Space Vehicle Design Criteria (Environment): Meteoroid Environment Model - 1969 [Near Earth to Lunar Surface]", 1969.

NASA SP-8038, "NASA Space Vehicle Design Criteria (Environment): Meteoroid Environment Model - 1970 [Interplanetary and Planetary]", 1970.

NASA SP-8042, "NASA Space Vehicle Design Criteria (Structures): Meteoroid", 1970.

Rawer, K. (Chairman), "International Reference Ionosphere--IRI 79 (edited by J.V. Lincoln and R.O. Conkright), World Data Center A, Rpt. UAG-82, Nov., 1981

Rawer, K., and P.A. Bradley, editors, "International Reference Ionosphere-Status 1986/87", The Committee on Space Research, Pergamon Press, New York, 1987.

Reagan, J.B., W.L. Imhof, and V.F. Moughan, "Characteristics of the August 1972 Solar Particle Events as Observed over the Earth's Polar Caps", in Collected Data Reports on August 1972 Solar-Terrestrial Events, Report UAS-28, Part III, World Data Center A for Solar-Terrestrial Physics, Boulder, CO., p. 676, July, 1973.

Sawyer, D.M., and J.I. Vette, "AP8 Trapped Proton Environment for Solar Maximum and Solar Minimum", NSSDC 76-06, NASA-GSFC, 1976.

Slowey, J.W., "Dynamic Model of the Earth's Upper Atmosphere", NASA Contractor Report 3835, Sept., 1984.

Tsao, C.H., R. Silberberg, J.H. Adams, and J.R. Letaw, "Cosmic Ray Effects on Microelectronics, Part III: Propagation of Cosmic Rays in the Atmosphere", NRL Memorandum Report 5402, Naval Research Laboratory, Washington, DC 20375, 1984.

Vampola, A., "Solar Cycle Effects on Trapped Energetic Particles", to appear in J. SPACECRAFT, 1989.

Vette, J.I., et al., "AE8 Model for Inner and Outer Zone Electrons at Solar Minimum and Solar Maximum: Unpublished electron model replacing AE5, AE6, and AE17" (see NSSDC reports 72-06, 72-13 (AE-4), 72-10 (AE5), and 76-04 (AE6), 1986.

Whitten, R.C., and I.G. Poppoff, "Fundamentals of Aeronomy", J. Wiley & Sons, Inc., New York, 1971.

FIGURE CAPTIONS

1. Neutral atmosphere vertical density profile in units of n^o/cm^{-3} (adapted from Carrigan and Strivanek [1974]).

2. Polar view of the northern hemisphere. Polar coordinates are employed such that the radial distance is in intervals of equal latitude (0° is the equator) while the angular coordinate is east longitude. Neutral atmosphere conditions for K_p=6, $F_{10.7}$=220, Day No°=357.5, and height=400 km as computed by the Jacchia 1972 model are shown.

3. Neutral temperature for the Jacchia 1972 model in K (same conditions as Fig. 2).

4. Number density of Oxygen for the Jacchia 1972 model in no^o/cm^{-3} (same conditions as Fig. 2).

5. Northern hemisphere view of percentage deviation of neutral density (N_2) and temperature from the Jacchia 1972 model for a geomagnetic storm of K_p=6$_o$ (after Slowey [1984]). "ST" means storm conditions; "QT" means normal Jacchia 1972 prediction.

6. Meridional cross section of the Earth's magnetosphere showing the dominate plasma regions and the magnetic field configuration for a noon-midnight cut.

7. Polar view of the northern hemisphere at 400 km as in Fig. 2 for the total magnetic field amplitude as predicted by the POGO magnetic field model. Units are G.

8. Polar view of the northern hemisphere as in Fig. 2 for the absolute v±B induced electric field at 400 km. The POGO magnetic field and a 90° orbital inclination were assumed.

9. Solar photon flux density spectrum at the Earth (after Grard [1973]).

10. The Zurich smoothed sunspot number variation between 1950 and 1988. Superimposed on the figure are the fluences for the individual large solar flare proton events during the same period (from Daly [1988]).

11. Structure of the Solar Wind magnetic field background and that associated with a high speed solar stream showing the effects of compression and shock. Also shown is the hypothesized termination shape for the Solar Wind beyond the orbit of Pluto.

12. Representative midlatitude daytime and nightime electron density profiles (top) and temperature profiles (bottom) for sunspot maximum (———) and sunspot minimum (- - - - -); Carrigan and Strivanek [1974].

13. Representative midlatitude daytime and nightime ion composition profiles for daytime (top) and nighttime (bottom) for sunspot minimum; Carrigan and Strivanek [1974].

14. Northern hemisphere view (coordinates as in Fig. 2) of the electron environment at 400 km for a sunspot number of 100 as predicted by the IRI model. a) electron density; b) electron temperature.

15. Representative differential electron flux spectrum for an auroral arc (Gussenhoven et al. [1985]).

16. Constant flux contours (after Daly [1988]) for protons and electrons as predicted by the AE8/AP8 radiation models (Vette et al. [1986]; Sawyer and Vette [1976]).

17. Comparison between relative abundances of the elements from hydrogen to the iron group normalized to carbon (C=100) for Cosmic Rays (heavy line) and for the Solar System (light line). From Meyer et al. [1974].

18. The energy spectra of Cosmic Ray protons (line) and electrons (points) as measured near the Earth. From Meyer et al. [1974].

19. Comparison of the August 1972 solar flare proton event fluxes with the Galactic Cosmic Ray fluxes for 1972 (Reagan et al. [1973]).

20. Comparison between the predicted debris flux from Kessler et al. [1989] and the natural meteoroid flux from NASA SP8013 at 500 km.

21. Cometary meteor velocity distribution as measured by different groups at 1 AU (from Morgan et al. [1988]).

22. Debris flux as a function of altitude as measured by the U.S. Space Command and as predicted by the debris model (Kessler et al. [1989]) for 10 cm objects in January 1987. An orbital inclination of 60° is assumed. From Kessler et al. [1989].

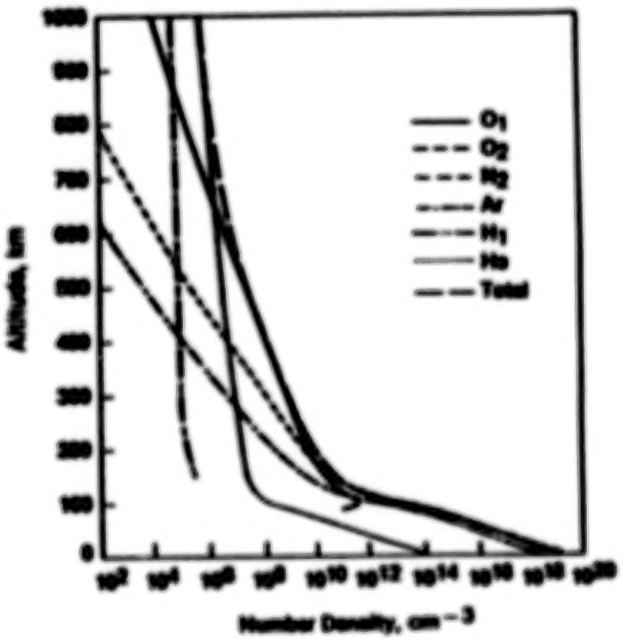


FIGURE 1.



FIGURE 2

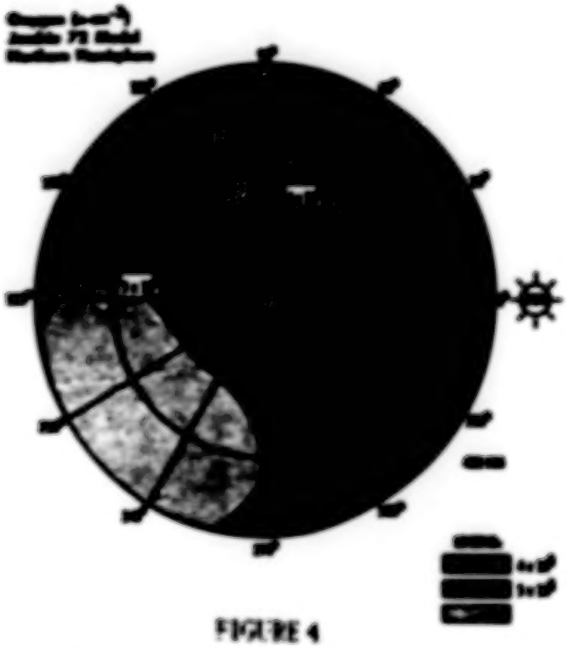


FIGURE 4

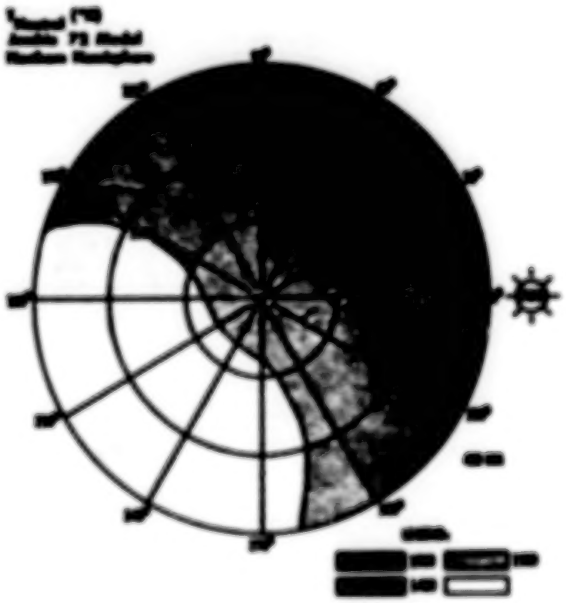


FIGURE 3





FIGURE 5



FIGURE 6.

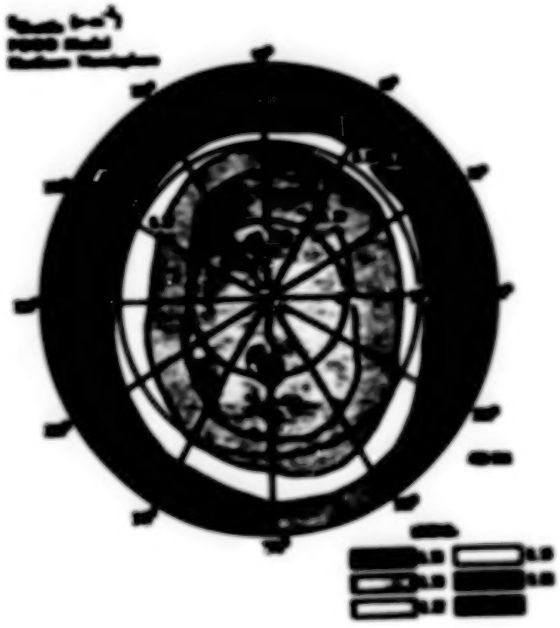


FIGURE 8.



FIGURE 7

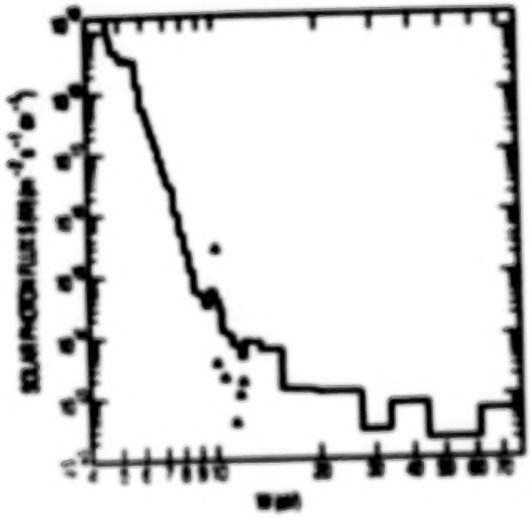


FIGURE 9

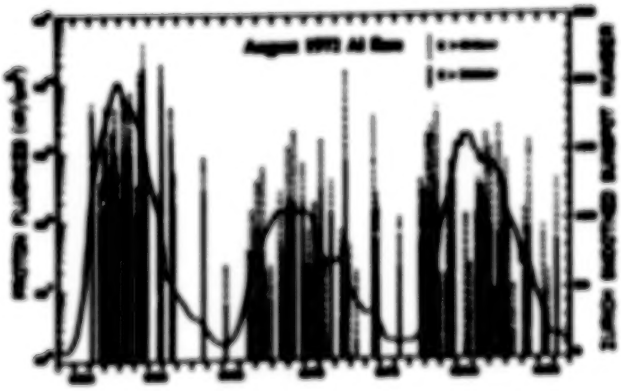
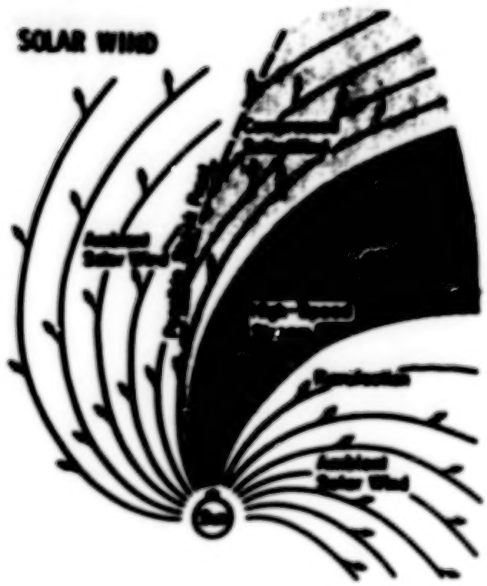


FIGURE 10.



MAGNETIC HELIOSPHERE

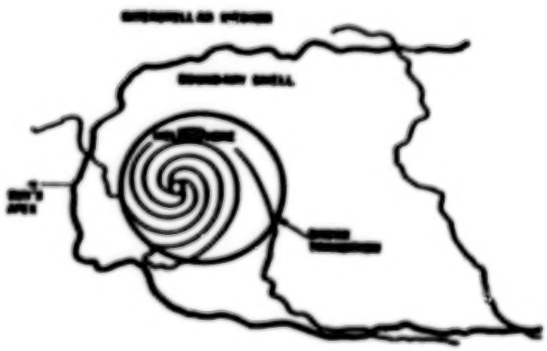


FIGURE 11.

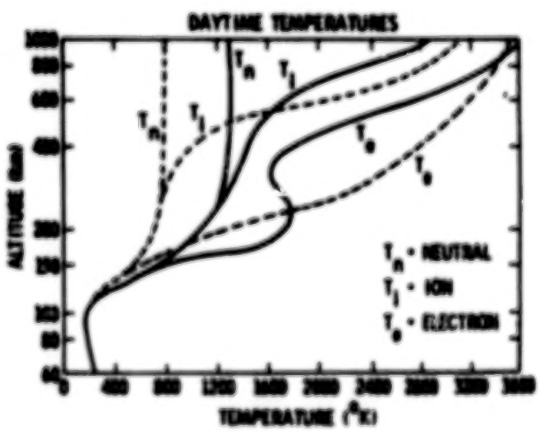


FIGURE 12A.

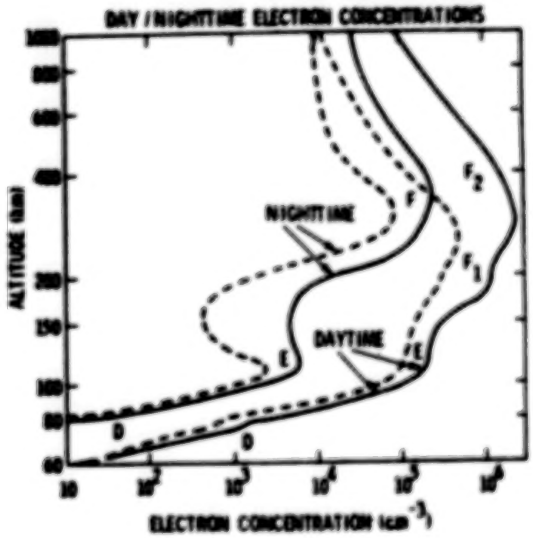
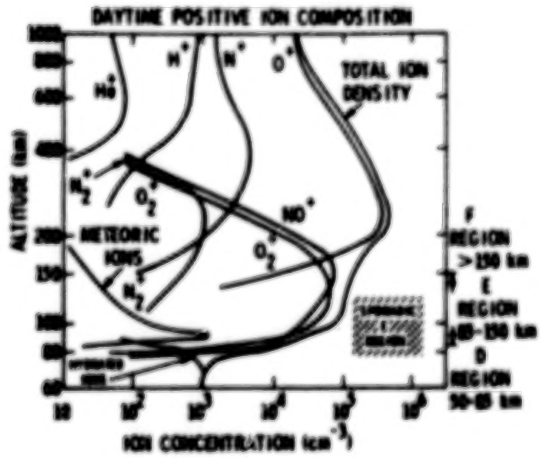


FIGURE 12B



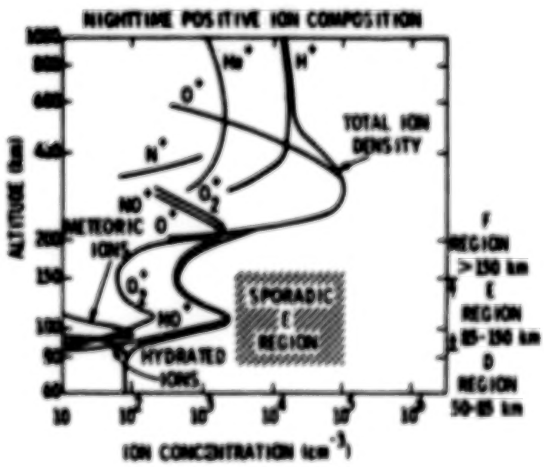


FIGURE 13

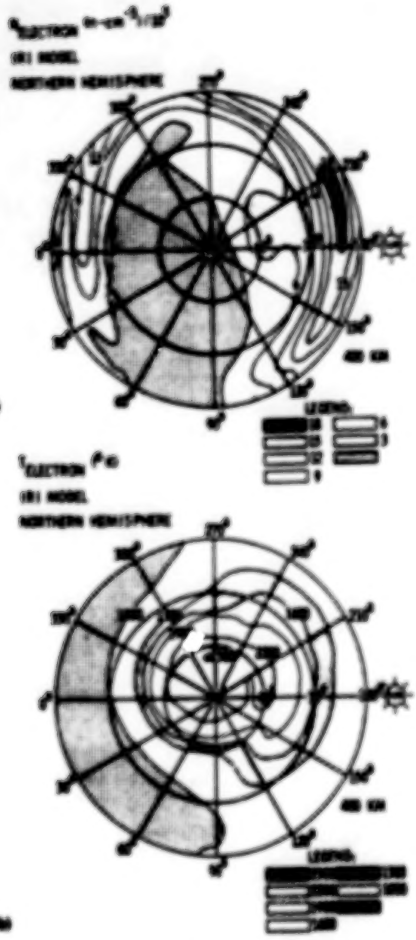


FIGURE 14

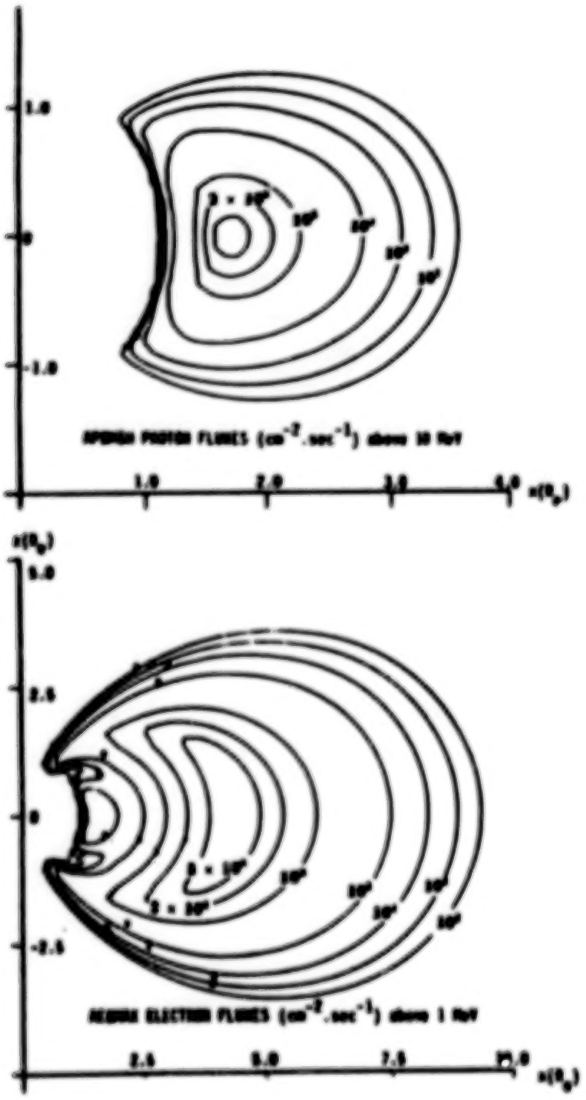


FIGURE 16

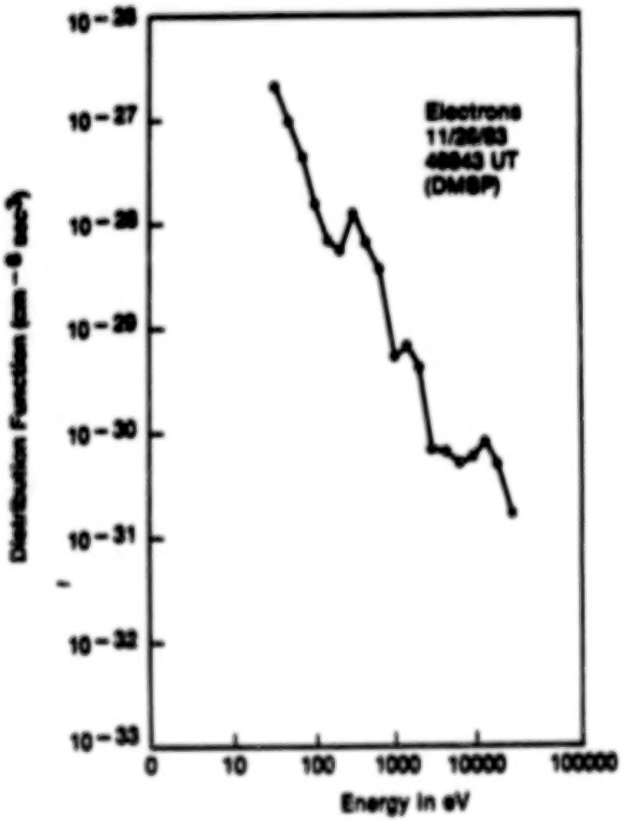


FIGURE 15.

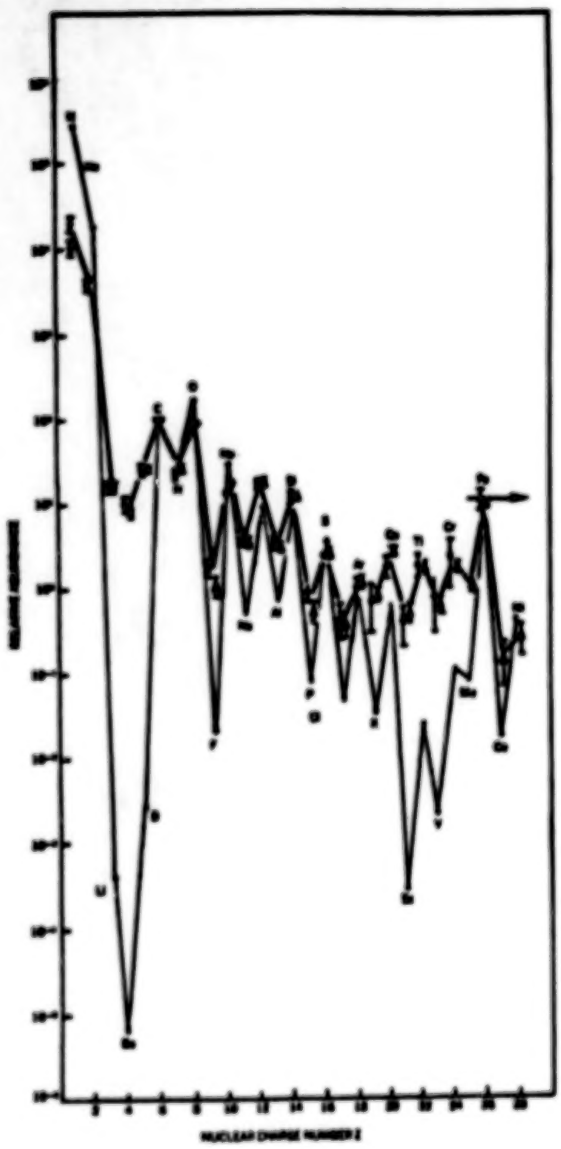


FIGURE 17

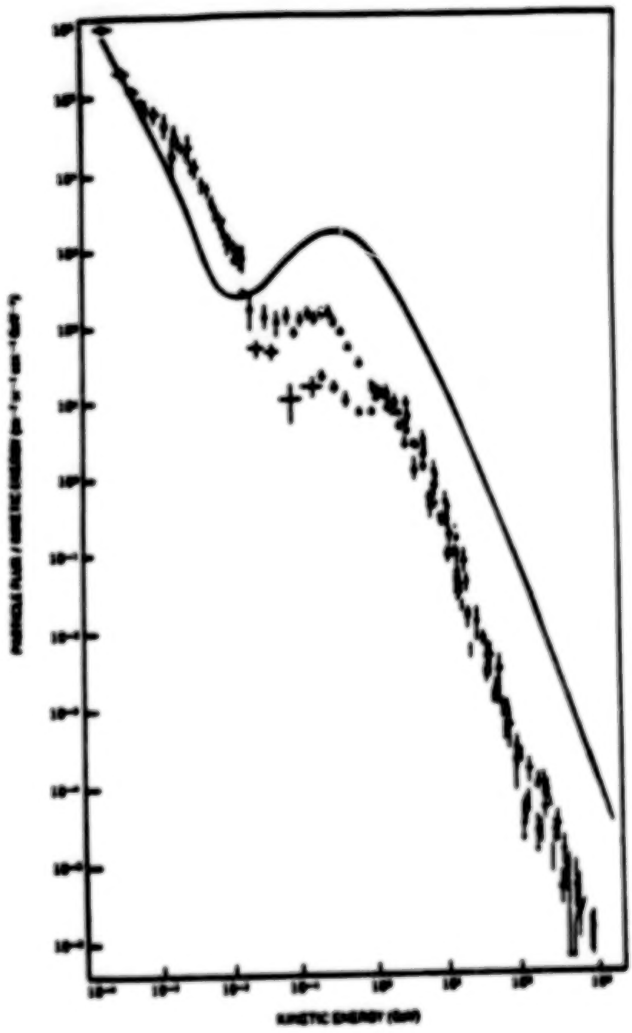


FIGURE 18

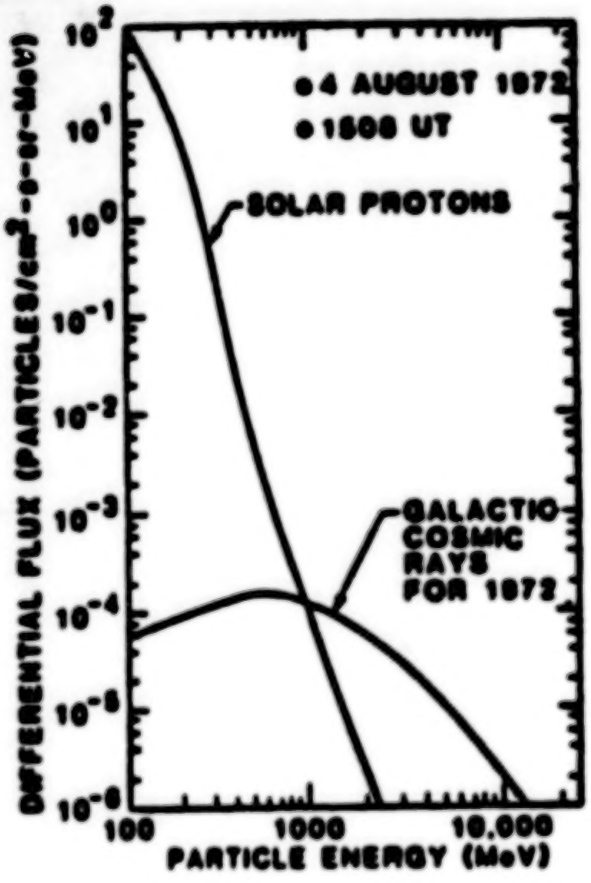


FIGURE 19

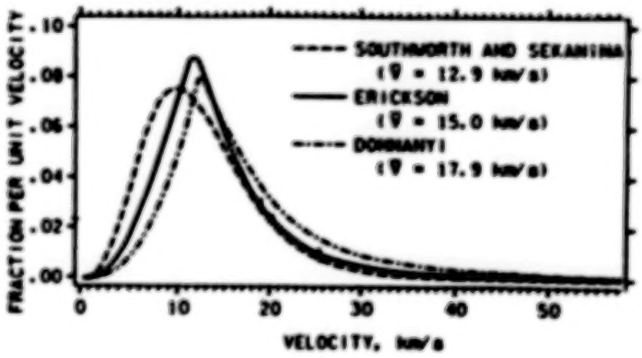


FIGURE 21

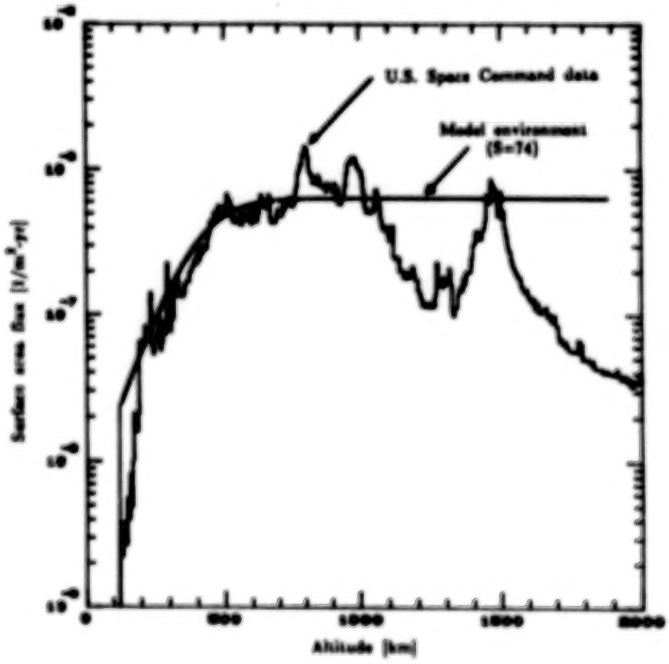


FIGURE 22

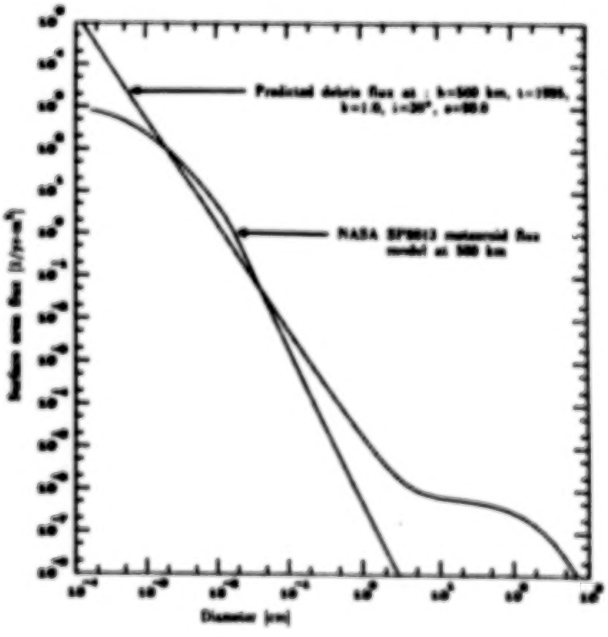


FIGURE 20

ENVIRONMENTAL CAUSED SPACECRAFT ANOMALIES IN ORBIT

J. Allen, D. Wilkenson, and H. Kroehe
NOAA/National Geophysical Data Center

(Paper not provided by publication date.)

DISCHARGE TRANSIENT COUPLING IN LARGE SPACE POWER SYSTEMS

By

N. John Stevens and R. P. Stillwell

TRW Space and Technology Group
One Space Park
Redondo Beach, CA 90278

ABSTRACT

Experiments have shown that plasma environments can induce discharges in solar arrays. These plasmas simulate the environments found in low earth orbits where current plans call for operation of very large power systems. The discharges could be large enough to couple into the power system and possibly disrupt operations. In this paper, the general concepts of the discharge mechanism and the techniques of coupling are discussed. Data from both ground and flight experiments are reviewed to obtain an expected basis for the interactions. These concepts were applied to the Space Station solar array and distribution system as an example of the large space power system. The effect of discharges was found to be a function of the discharge site. For most sites in the array discharges would not seriously impact performance. One location at the negative end of the array was identified as a position where discharges could couple to charge stored in system capacitors. This latter case could impact performance.

INTRODUCTION

The space environment can induce discharges in space power systems either by surface charging via geomagnetic substorm environments in geosynchronous orbit or by interactions between biased surfaces and the space plasma environment in low earth orbits [1,2]. These discharges have been demonstrated in laboratories and there is sufficient space flight data to substantiate the laboratory results [3,4,5,6]. It is important, therefore, to know how large space power systems will respond to these transients. While the phenomenon in both orbits is interesting, this paper will address only the low orbit, large power systems.

The tests indicated two major results. The first was that, as the bias voltage increased above 100 volts, there was a dramatic increase in collected current (see Figure 4). This increase was plasma density dependent - larger currents were collected for higher plasma densities. This effect was explained by the fact that the interconnect electric field extended over the cover glass and accelerated plasma particles into the cover glass increasing secondary emission. These secondary particles were then collected by the interconnects increasing the total plasma collected current. This effect was called "snap-over' [13].

The second result was the discharges that occurred when the negative bias voltage, relative to the plasma potential, exceeded a density-dependant threshold (see Figure 5). These discharges could shut off laboratory power supplies [3].

Ground tests were run on a continuing basis with the available technology solar cells (2 X 2, 2 X 4, 6 X 6), using standard and wrap-around interconnects. The segments tested ranged from small 100 cm^2 arrays to substantial 2 m^2 arrays. Testing with both bias power supplies and self-generated voltage arrays was conducted. The results were similar: snap-over at positive voltages and discharges at negative voltages.

There were two flight experiments conducted to measure the interaction in actual space environments: Plasma Interaction Experiment (PIX) 1 and 2 [14,15]. The results here were the same as in the ground simulation tests. The discharge data for both ground and space results have been assembled and, while there is some uncertainty in the absolute threshold voltage values, there is no doubt that discharges do occur [7,16]. For plasma densities corresponding to the Space Station orbit, discharge threshold values range from -138 to -250 volts. The low value is from the self-generated voltage testing [17].

The largest space power system planned for low earth orbit operations is the Space Station, Freedom. This Station is to have an array that will generate 215 kw of power at a nominal voltage of 160 volts in order to provide the 70 kw required at the Station electrical loads [7]. The final operating Station is planned to have a solar array configured in eight separate

blankets in four wings. An artist's conception of one version of the final operating configuration of the Station is shown in Figure 1 and gives an idea of its physical size. If there were discharges in these arrays, then there could be system damage if the discharge could drain the array power.

The current plan for the Station is to have the power generated in the solar arrays and transmitted to the Station electrical loads via an AC transmission line. The option of using a DC system from the arrays to the loads is still being discussed. The array configuration consists of 16 solar cell sectors, connected in series (with a bypass diode), to form a 160 volt block. The blocks are then connected in parallel to form the 13.5 kw blanket. There are two blankets per wing (see Figure 2). The solar cell chosen for the Station is a new 8 X 8 cm silicon cell.

Before considering the interactions between the Space Station and the environment in more detail, it is necessary to review the background data on plasma induced discharges in solar cells.

DISCHARGES IN SOLAR CELLS

The behavior of solar cells, biased to various voltages in plasma environments, has been studied since the late sixties [3-6, 8-12]. These experiments used small segments of arrays, biased by external power supplies, to measure the plasma-cell interaction. The test arrangement is similar to that shown in Figure 3. The initial concern was for possible power losses that could be induced in the array because of the parasitic parallel loop through the plasma. The tests soon showed that there were other concerns.

The missing piece of data that is required to assess the Station behavior is data on the 8 X 8 cm solar cell. At this time, there has been no plasma interaction testing of these cells. There are differences in the construction techniques for these new cells and these may cause the cell to respond differently to plasmas. Since this data is missing, it will be assumed, for the following discussion, that they will behave similarly to the other cells.

APPLICATION TO SPACE STATION POWER SYSTEM

A space power system, operating at a given voltage, will interact with the space plasma environment such that the net current collected from the plasma is zero. This means that the structure potential will vary, relative to the space plasma potential, until this current balance is reached. Since electrons are more mobile than ions, this usually means that the system will float negative to repel electrons and attractions.

The Space Station design currently calls for the positive side of the array to be grounded to the structure. The structure will then be at or near plasma potential and the array voltage will be negative. There will be velocity effects as the Station moves around its orbit and plasma conditions change [7]. These effects are illustrated in Figure 6. Previous studies of system behavior has indicated that the voltages will not be high enough for snap-over to occur unless active charge control techniques are used. This means that discharges would be the dominant concern for the Station power system performance. As shown in Figure 6, all blocks in the eight blankets have areas that are at sufficiently negative potential where discharges can occur. The discharge threshold used is based on the minimum values found in the PIX-2 data extrapolated to Space Station altitudes.

The actual discharge transient is not well characterized. Measurements have been attempted that appear to show pulses of several microsecond duration, but this is more probable the test sample response than a actual discharge [18]. Other experiments have shown that the discharge is really a multiple process involving the initial discharge transient stretched by decharging of other capacitors in the array [19]. It is this latter type of discharge process that will be considered here.

The Station power system is still being designed so that only the conceptual elements of that system can be included in this discussion. The DC generation - DC transmission and the DC generation - AC transmission power distribution systems are shown conceptually in Figures 7 A and B. An initial study of this system was concerned primarily with the effect of discharges on the Station electrical loads [7]. It used a model in which the structure ground was firmly tied to space potential. The batteries, required to provide power during eclipse, were mounted on the wings. A solar array switching unit (SASU) was also included to maintain array power at the maximum power point by switching units in and out of the circuits as required. The SASU was simulated as a capacitor and resistor in this model. A diode characteristic was used to simulate the solar array performance.

In that study, discharges were simulated simply by turning off various levels of power for periods of up to 60 microseconds. Small discharges were shown to have little effect on the electrical load. This is because the batteries would come on line to maintain power. Any ripple due to the transient would be damped out by the transmission line inductance. For the DC-DC system appreciable power losses occurred only when the whole array was involved in a discharge or when the array diodes failed. This was due to the back-biased array being an additional load for the battery to supply. The DC-AC system didn't seem to respond to complete array shut down of up to 60 microseconds.

Based upon the assumptions used, it became apparent that discharge effects in the array should be evaluated in more detail. Therefore, this study was undertaken to evaluate the effect in the DC-DC system. Since the interest was now in the array, the circuit model was simplified as shown in Figure 8. In this model, the Station is now coupled to space with a typical value of 400 picofarads. The battery system is also diode protected from reverse current flows. This eliminates any possibility of having the battery being drained by the dark array during eclipse.

The characteristics of the discharge itself are not really important. The discharge is assumed to trigger a process resulting in the decharging of other capacitors. The process assumed is that a discharge occurs in the negative portions of the array locally reducing the voltage. This causes the array material capacitors (cover glass and substrate) to decharge by ejecting electrons to space to correspond to the change in voltage. This stretches out the duration of the pulse. For a small discharge involving two sectors, the cover glass and substrate capacitance would be on the order of 0.35 microfarads. The resistance to space has been set at 50 ohms and the discharge inductance computed at a value to give an underdamped pulse ($1X10^{-4}$ H).

This discharge must occur in the array in or between sectors where the voltage is sufficiently negative with respect to the space plasma potential. Since an absolute discharge threshold has not yet been established, two separate discharge locations will be considered in the following sections.

Discharge Occurring Between Sectors

A discharge is assumed to occur between the 25^{th} and 26^{th} sector where the voltage, relative to the space plasma potential, is -154 volts (switch position marked A in Figure 8). The discharge is triggered by a breakdown in the cell circuit. This drives the voltage at this point towards zero based on the relationship that the discharge current equals the discharge capacitance times the time dependent change in voltage. This change in voltage causes the charged cover glass and substrate capacitors to decharge stretching the discharge pulse. This type of discharge also will cause the sector voltages to change since the current flow in the cells has been modified. When the discharge current was less than the plasma current collected by the block, then the plasma currents drove the voltage distribution back initial value. The capacitor controlling this rate of change of voltage was assumed to be the system capacitance to space. Hence, the return to normal conditions is more rapid. This effect has been modeled with a circuit analyser code and the results shown in Figure 9. A single discharge, at this location, could effectively shut down the block or reduce the power generated by 1/36 of the total blanket power for about 60 sec (assuming 36 parallel blocks). The sector cell reverse current resistances and the sector diodes would prevent other blocks from being affected by the discharge. The battery would come on-line and maintain power to the load and the battery diodes would isolate both the battery and the Station loads from the discharge. The problem that could occur here would be stressing the diodes; although they should be capable of withstanding such stresses. Hence, this type of discharge should not cause any serious disruption of power service in the station. If multiple discharges do occur, then it is possible for several blocks in all the blankets to go down simultaneously. If the battery is charged, then these losses again would not affect the Station loads. There would still be a concern if the discharge rate were high, but such rates for large arrays have not yet been determined.

Discharge At End Of Block

This type of discharge is assumed to occur at switch position marked B in Figure 8 at breakdown voltage of -160 volts. The process involved in the discharge is the same as in the preceding case: a discharge triggers the decharging of the cover glass and substrate capacitors. The effect here, however, is far more serious. The discharge is triggered at the most negative voltage area of the block and again, the voltage is driven towards zero. The block current is available to add to the discharge as before. Now, the other blocks can add their current to the discharge since there are no blocking diodes nor high resistance cell paths to hinder current flow. This means that the SAEU capacitor can also decharge. This unit has a 2000 microfarad capacitor charged to 160 volts or a stored charge of 0.32 coulombs. The behavior during this type of discharge was also modelled and the results are shown in Figure 10. As shown the current in the pulse can rise 75 to amperes in 70 sec. This could be sufficient to cause significant damage to wires and components. While this is going on, power for the station electrical loads will be provided by the battery until this supply is exhausted.

If this type of discharge is so serious, why hasn't it been observed in the laboratory tests of array segments? The answer is that it has been. The early test results mentioned that the discharge would drain the full capability of the power supply. As the tests became more sophisticated, current limiting resistors were added in the test lines to isolate the power supply during the discharge so that the discharge process could be studied. This prevented current flow from the supply and since the array samples were small, the discharge pulse was small.

There are possible mitigation techniques that can be used here to prevent serious consequences from this type of discharge. This is to provide additional diode protection to prevent capacitor decharging current flows.

CONCLUDING REMARKS

Large solar array power systems must function in the space plasma environment for long periods of time. Possible interactions in such arrays have been studied for the past twenty years by biasing small segments of the array to various voltages while exposed to a simulated space plasma environment. Two major effects were found: the "snap-over" current collection when the bias voltages were greater than 100 volts and discharges that occurred in negative biased regions of the array. Snap-over phenomenon resulted in high plasma current collection which implied additional power losses. Discharges generated transients that could conceivably damage systems. Previous system evaluations have indicated that the proposed operating voltages would preclude operating at greater than 100 volts relative to the plasma potential, so this is not considered to be a serious problem. That leaves only the question of discharges.

The Space Station power system represents the largest power generating array to be flown on low earth orbit. This array would generate 215 kw of power while operating at 160 volts. Solar cells for this array are to be the new 8 X 8 cm cells for which there is no data on plasma interactions. Under the assumption that these cells would behave similar to all other types of cells, system evaluations have been conducted. An initial study concluded that there would be minor disruption in the power flow unless either the entire array shut down temporarily or that the back-bias diodes failed. Then there could be a 10% power loss for the duration of the discharge (up to 60 microseconds durations were considered).

In a recent study of solar array discharges, the results indicated that a discharge could couple into other capacitors in the system, stretching out and amplifying the pulse. This concept was applied to the Space Station system. The Station power distribution system designed used in the initial study was modified to incorporate space capacitance and discharges occurring in two different locations on the array were evaluated. If the discharge occurred at the end of a block, then the results were more serious. Such a discharge would shut down all of the blocks in a blanket. The SASU capacitance could also dissipate its charge appreciably increasing the current flow. A mitigation technique to minimize this effect is to use blocking diodes in the array lines between the blocks. A major unknown in these studies is the discharge repetition rate. If discharges are frequent, then they could prevent or reduce battery charging capability.

The basic assumption in this study is that the 8 X 8 cm cells behave the same as the other, older cells. If these new cells have the same discharge characteristics, then the effects described here will happen. If the cells behave in a different manner, then the whole question would have to be revisited.

REFERENCES

1. Archer, J. S., Rauschenbach, H. S. and Stevens, N. J.: "Investigation of ESD Hazard for Large Space Solar Arrays Configured with GFRP/Kapton Substrates," AIAA Paper 89-0617, January 1989.

2. Stevens, N. John: "Interactions Between Large Space Power Systems and Low-Earth Orbit Plasmas," Spacecraft Environmental Interactions Technology - 1983, NASA CP-2359/AFGL-TR-85-0018, 1985, pp. 253-276.

3. Ramsrud, K. L.: "High Voltage Solar Array Experiments," NASA CR-121280, 1974.

4. Herron, B. G., Bayless, J. R. and Worden, J. D.: "High Voltage Solar Array Technology," AIAA Paper 72-443, April 1972.

5. Domitz, S. and Kolecki, J.: "Effect of Plasma Currents on Solar Array Power Output," Spacecraft Charging Technology - 1978 - NASA CP-2071/AFGL-TR-79-0082, 1979, pp. 359-375.

6. Stevens, N. John: "Solar Array Experiments on the SHUTTLE Satellite," NASA TMX-71458, 1973.

7. Stevens, N. J. et.al.: "High Voltage System Performance in Low Earth Orbit Plasma Environment Vol. 1 - Space Station Environmental Interactions," TRW 44870-6005-UT-00, NASA CR-179640, October 1986.

8. Inouye, G. T. and Sellen, J. M., Jr.: "TIROS Solar Array Arc Discharge Tests," Spacecraft Charging Technology - 1978 - op. cit, pp. 834-852.

9. Brady, G. F., Jr., Vance, D. A. & Greenberg, S. A.: "Charging & Discharging Characteristic of a Rigid Solar Array," Spacecraft Charging Technology - 1980, NASA CP-2182/AFGL-TR-81-0270, 1981, pp. 229-236.

10. Beaver, Renate & Staghus, J.: "Tank Testing of a 2500 cm² Solar Panel," ibid. pp. 211-227.

11. Grier, N. T.: "Experimental Results on Plasma Interactions with Large Surfaces at High Voltage," NASA TM-81423, January 1980.

12. McCoy, J. E. and Konradi, A.: "Sheath Effects Observed on a 10-Meter High Voltage Panel in Simulated Low Earth Orbit Plasmas," Spacecraft Charging Technology - 1978, op. cit., pp. 319-340.

13. Purvis C. K., Stevens, N. J. and Berkopec, F. D.: Interactions of Large, High-Power Systems with Operational Orbit Charged-Particle Environments," NASA TMX-73867, 1977.

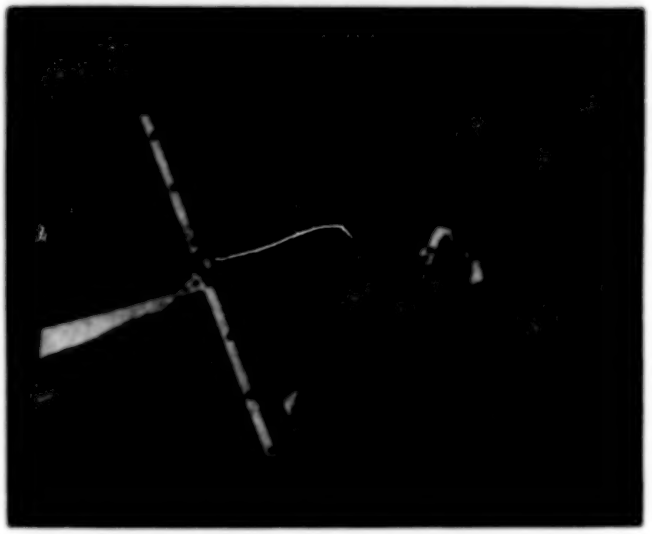
14. Grier, N. T. and Stevens, N. J.: "Plasma Interactions Experiment (PIX) Satellite Results," Spacecraft Charging Technology - 1978, op. cit, pp. 295-314.

15. Purvis, C. k.: "The PIX-II Experiment: An Overview," Spacecrft Environmental Interactions Technology - 1983, op. cit., pp. 321-332.

16. Ferguson, D. C.: "The Voltage Threshold for Arcing for Solar Cells in LEO - Flight and Ground Test Results," NASA TM-87259, March 1986.

17. Snyder, D. B.: "Discharges on a Negatively Biased Solar Cell Array in a Charged-Particle Environment," Spacecraft Environmental Interactions Technology - 1983, op. cit., pp. 379-388.

18. Stevens, N. J. and Stillwell, R. P.: "Environmentally Induced Discharges in Solar Array," paper presented at this conference.



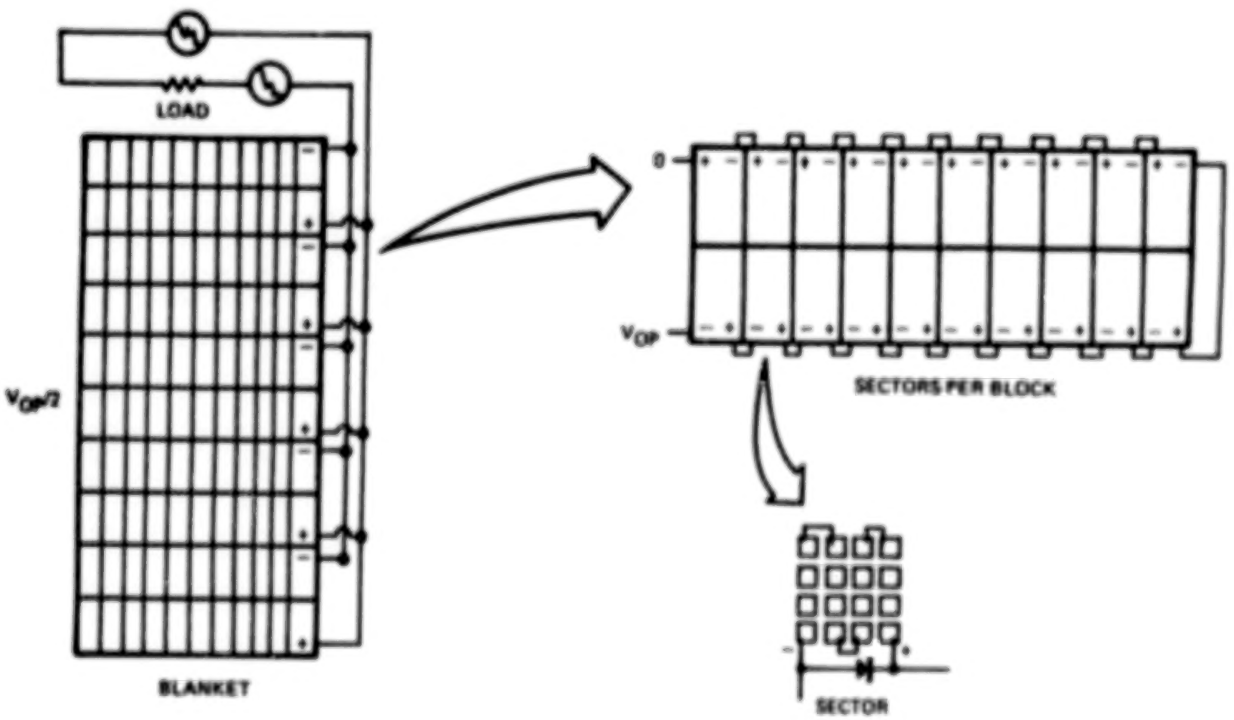


FIGURE 2. SOLAR ARRAY BLANKET CONFIGURATION

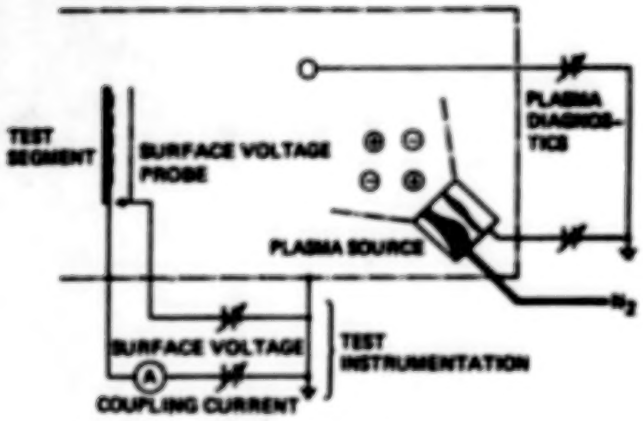
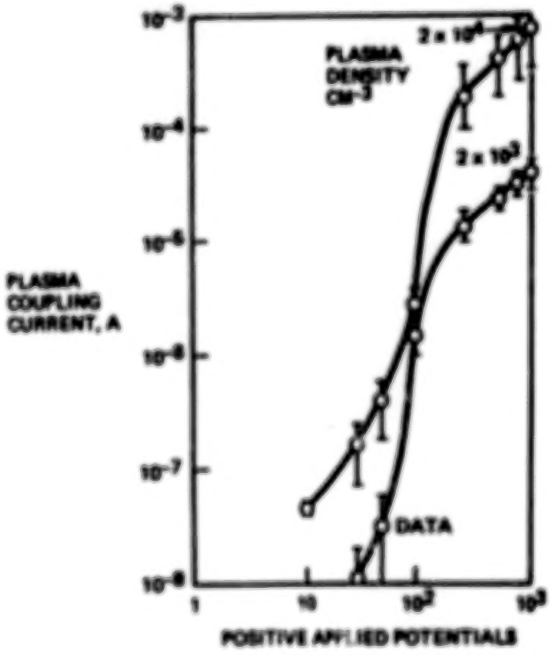
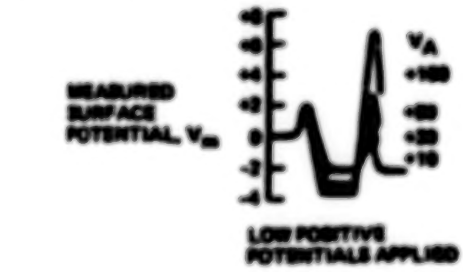


FIGURE 3. SCHEMATIC DIAGRAM OF TEST ARRANGEMENT

A. CURRENTS COLLECTED



B. SURFACE VOLTAGE PROFILES



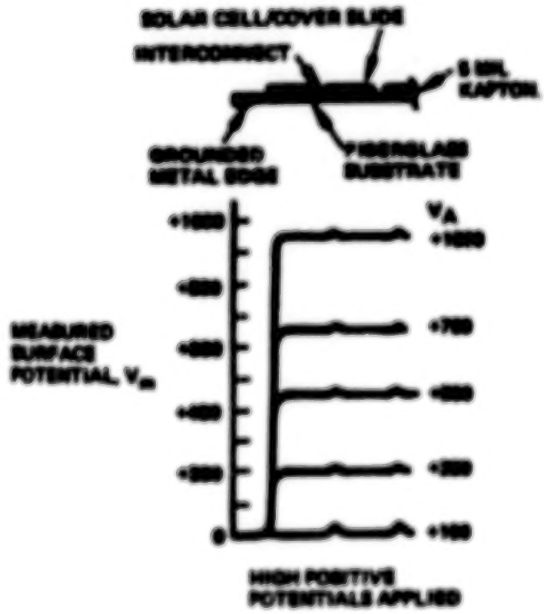
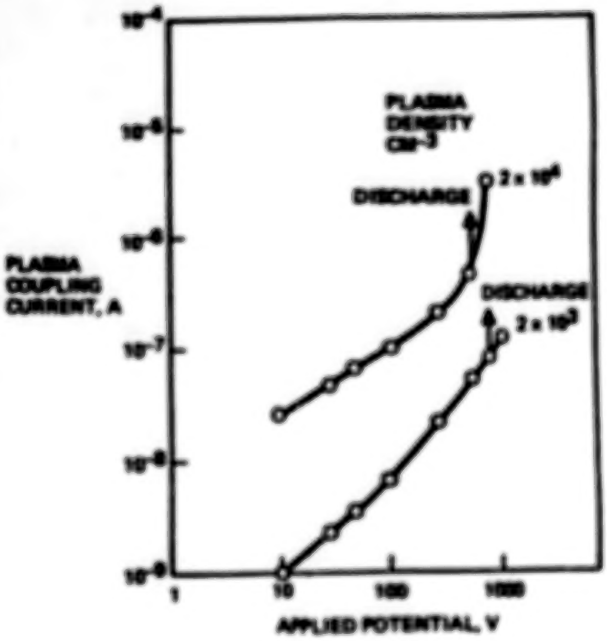


FIGURE 4. GROUND TEST RESULTS - POSITIVE BIAS

A. CURRENT COLLECTION



B. SURFACE VOLTAGE PROFILES

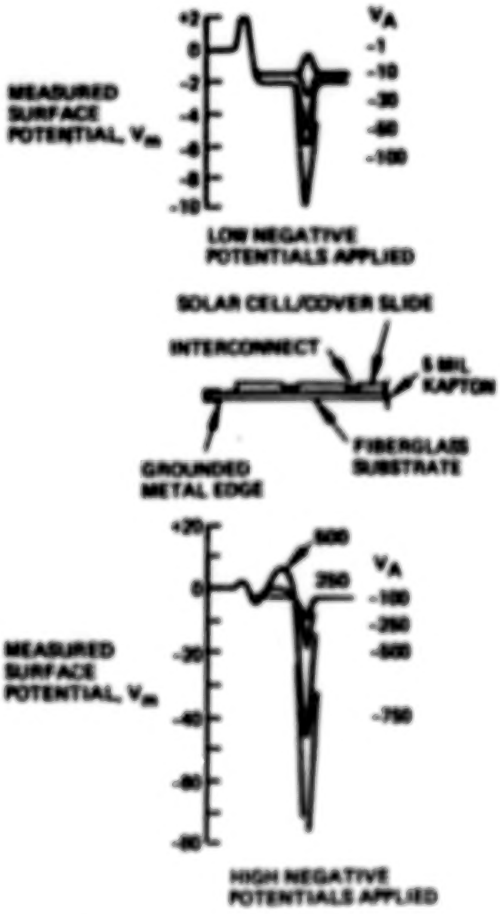


FIGURE 5. GROUND TEST RESULTS - NEGATIVE BIAS

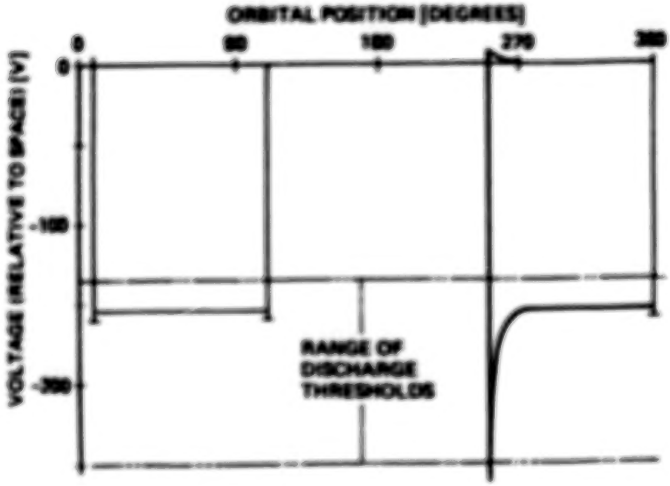


FIGURE 6. PREDICTED PERFORMANCE OF SPACE STATION ARRAY

A. DC GENERATION/DC TRANSMISSION

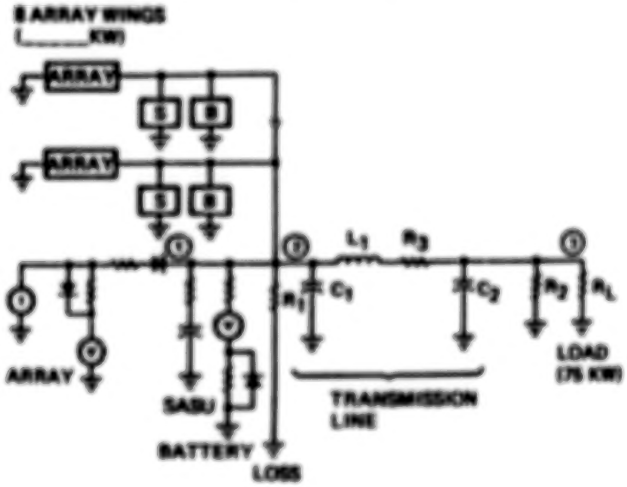


FIGURE 7. TRANSIENT RESPONSE CIRCUIT

B. DC GENERATION/AC TRANSMISSION, 20 kHz

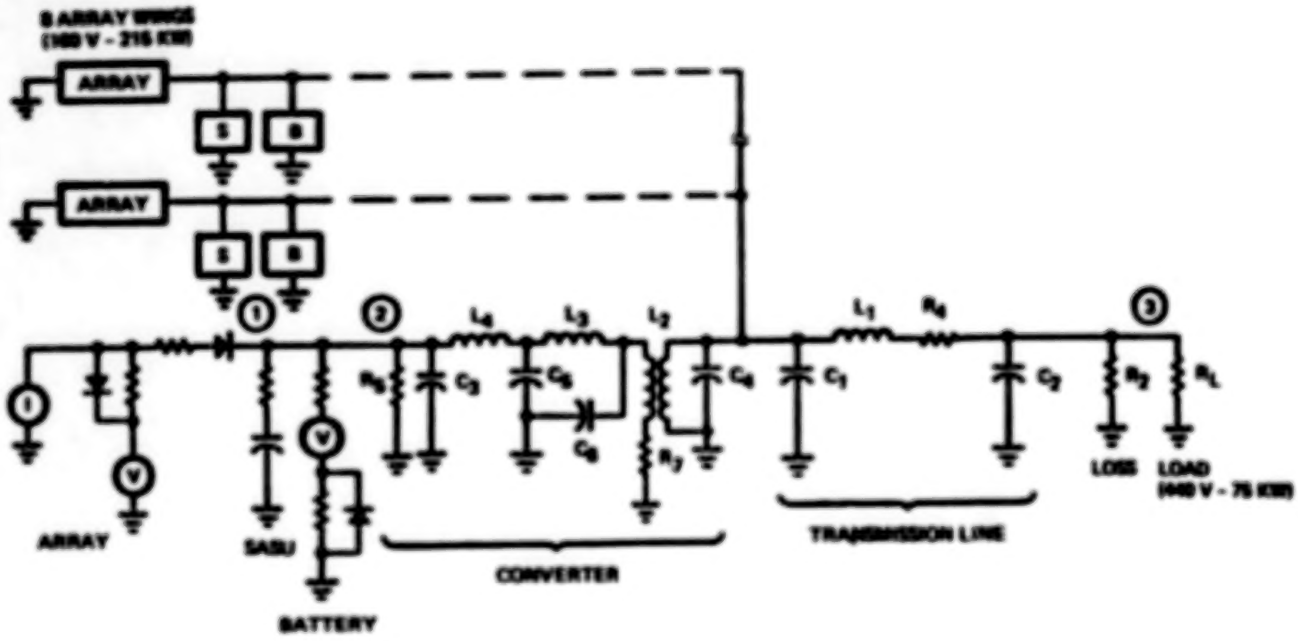


FIGURE 7. TRANSIENT RESPONSE CIRCUIT (CONTINUED)

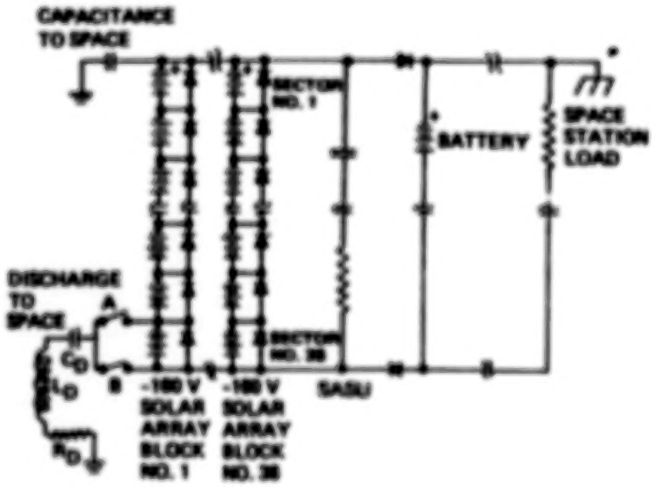


FIGURE 8. SIMPLIFIED SPACE STATION SOLAR ARRAY POWER SYSTEM SCHEMATIC

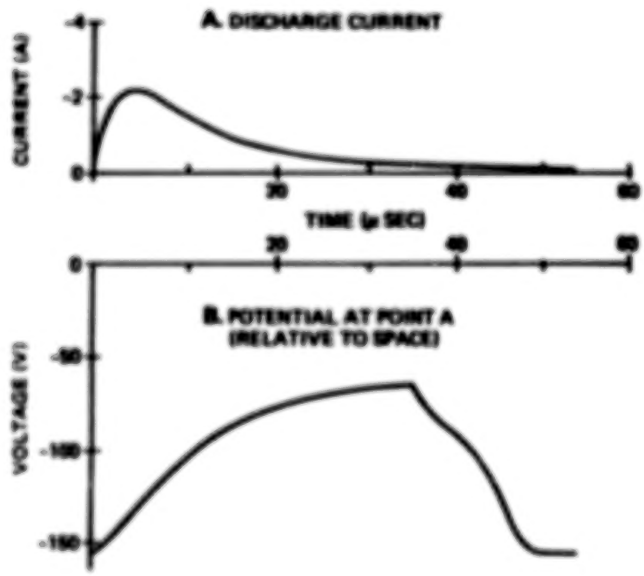


FIGURE 9. PREDICTED PERFORMANCE FOR DISCHARGE AT POINT A

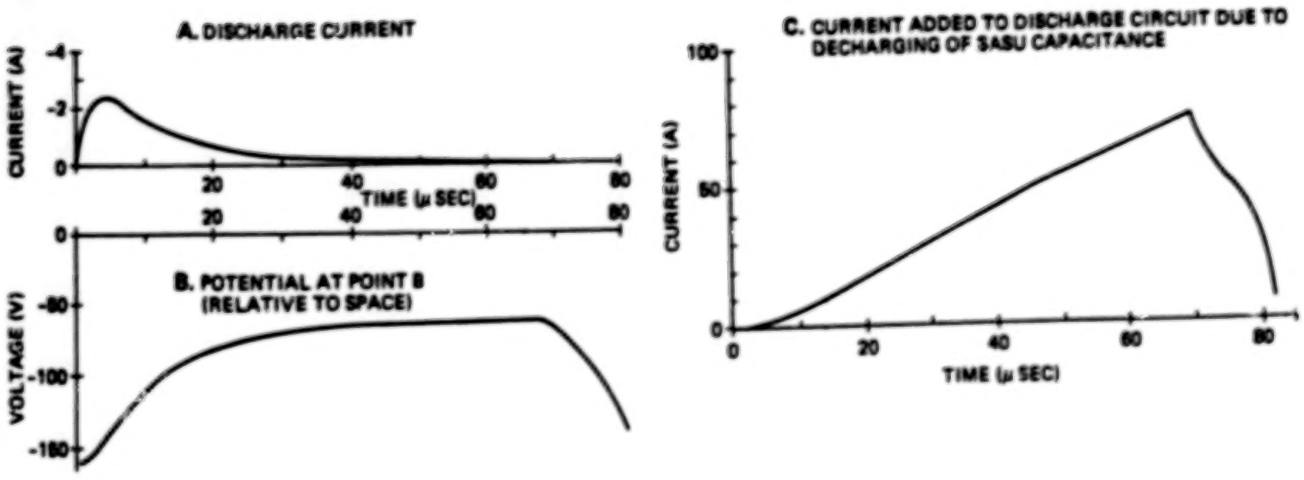
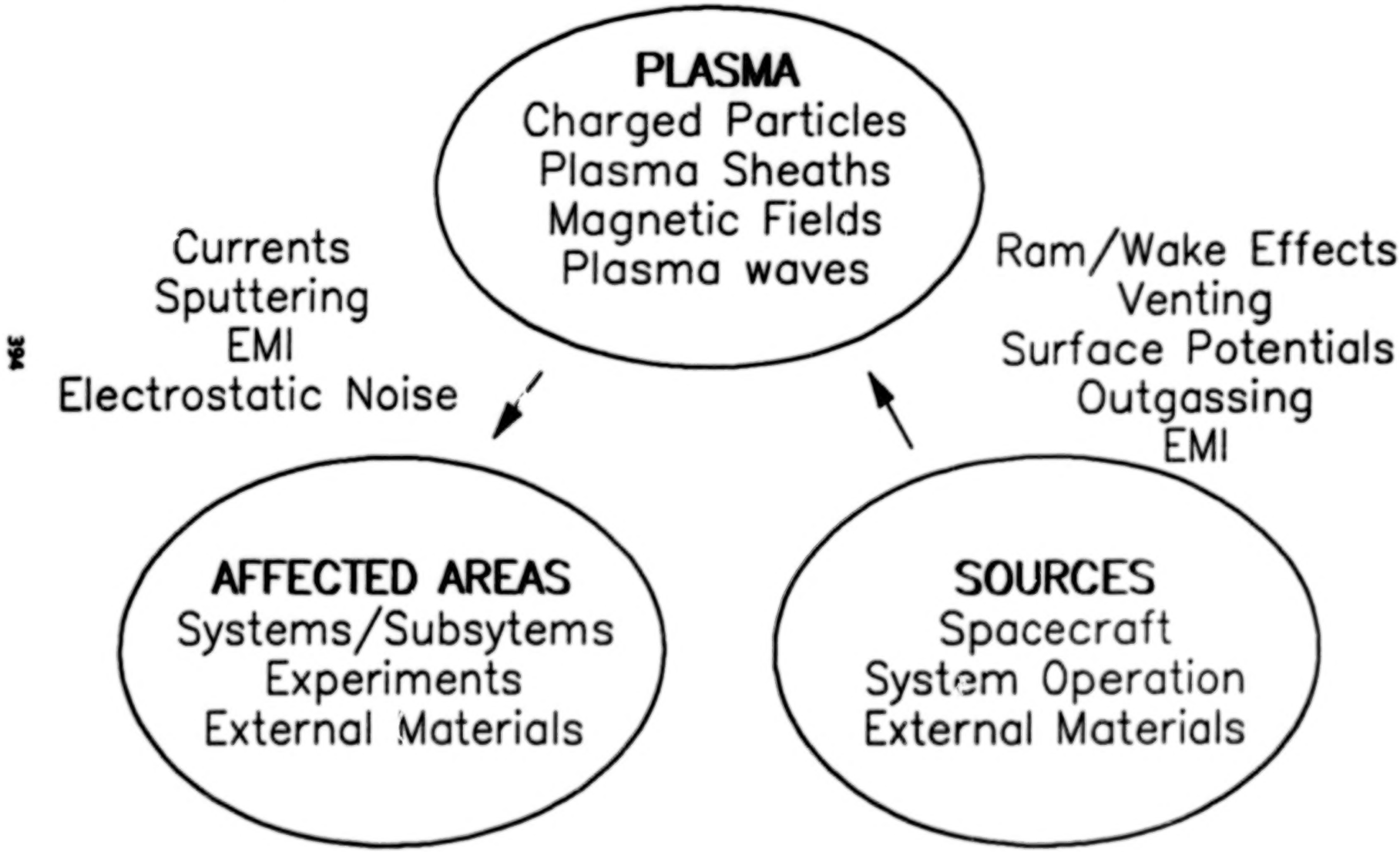


FIGURE 10. PREDICTED PERFORMANCE FOR DISCHARGE AT POINT B

Plasma Interactions and Effects for Large Systems

David B. Snyder
NASA Lewis, Cleveland, OH

PLASMA–SPACECRAFT INTERACTIONS



PLASMA ISSUES

Plasma tends to 'ground' surfaces.
Charging not as severe an issue. Fewer high energy charged particles than in GEO. (Except in Polar orbits)

But:
Current Collection is now an issue.
(Currents through structure, instrumentation)

Discharges on exposed, High Voltage Surfaces.

Floating Potential of Spacecraft.
(What should be used for reference ground?)
rapid fluctuations may induce structure currents

Electrostatic Noise. (Ram/Wake, venting, thruster firing)
Plasma supports wave propagation
Magnetic Field Guides waves

Multi-purpose vehicle.
When will systems/users interfere with each other?

Design Guidelines exist for Geo environment but not for LEO

NASA TP2361 Purvis, et al. (1984)

GEO Environment: low charge density, high energy particles

GEO Issues: Dielectric Charging, Arcing

GEO Recommendations: Maximize dielectric Conductivity, conductors must be grounded

PLASMA CHARACTERISTICS
(200 - 800 km altitude along equator)

day density: 10^5 to 10^6 #/cm^3

Temperature
electrons: 1000 to 2500 K (0.1-0.2 eV)
ions: 700 to 1900 K
Debye length ~ 0.5 cm
Thermal Current Flux
electrons: 0.004 A/m^2 ions: 2×10^{-5} A/m^2

night density: 10^4 to 5×10^4 #/cm^3

Temperature
electrons: 900 to 1100 K (0.1 eV)
ions: 700 to 1100 K
Debye length ~ 2 cm
Thermal Current Flux
electrons: 2×10^{-4} A/m^2 ions: 2×10^{-5} A/m^2

Magnetic Field: 2 to 4 $\times 10^{-5}$ tesla

Magnetic Field (vxB-l)

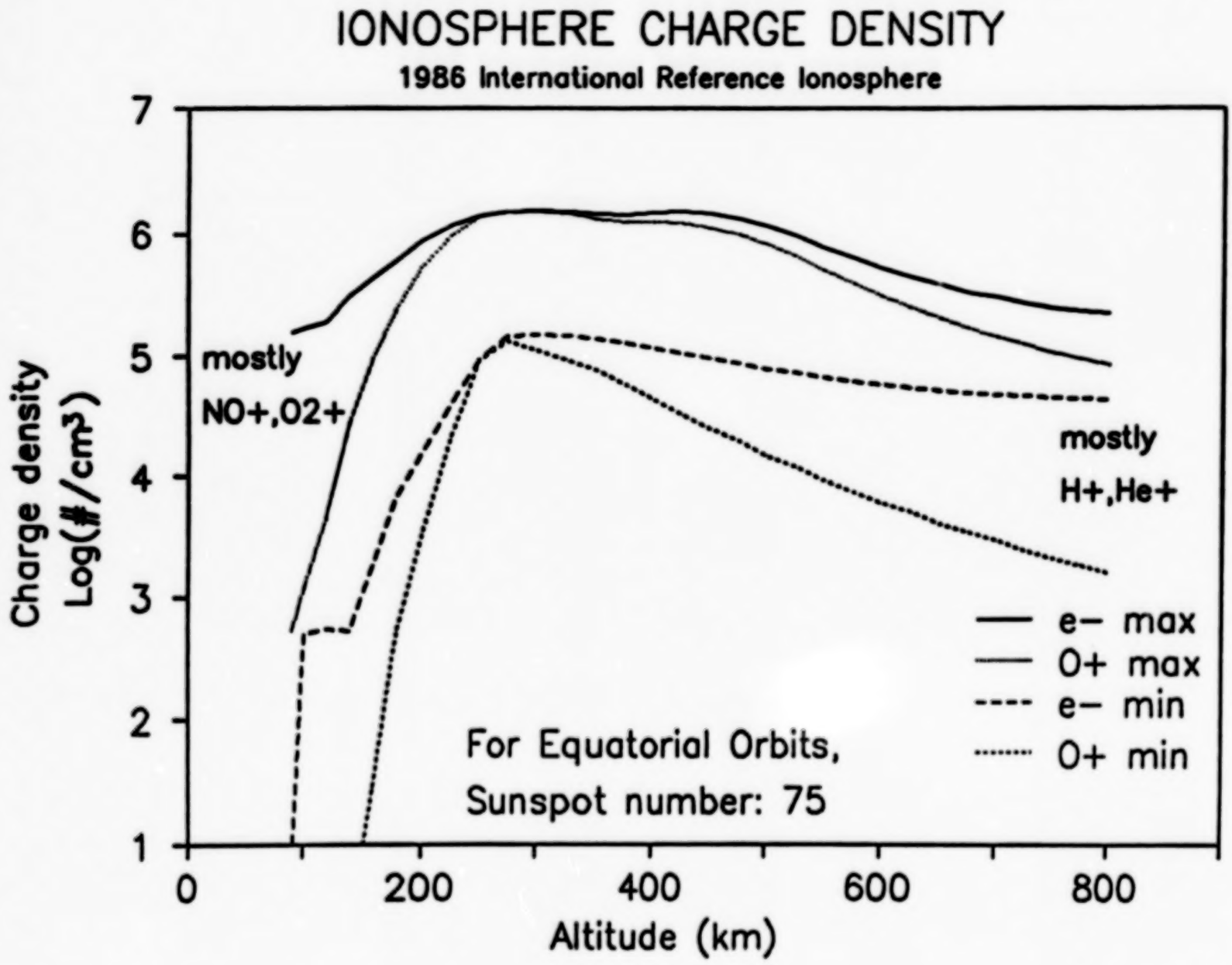
at 400 km: |B|=3.4E-5 tesla (an approximate high field)
20 deg dip angle

orbit velocity: 7.67 km/s

E = v x B => .26 V/m

Space Station Dimensions (approximate):
length: 110 m (10 V) (20 degree dip angle)
height (across solar arrays): 70 m (18 V , 0 deg. dip)

Chemical interactions with ions may be important.
Structure currents induced.
Plasma currents radiate noise.


IONOSPHERE CHARGE DENSITY
1986 International Reference Ionosphere
Charge density
Log(#/cm³)
7
6
5
4
3
2
1
0
200
400
600
800
Altitude (km)
mostly
NO+,02+
mostly
H+,He+
e− max
O+ max
e− min
O+ min
For Equatorial Orbits,
Sunspot number: 75

SPACECRAFT FLOATING POTENTIAL

Steady State when:
ion current = electron current

This is achieved locally by charging a few kT negative to reduce the thermal electron current.

<u>Biased systems</u> (exposed conductors)

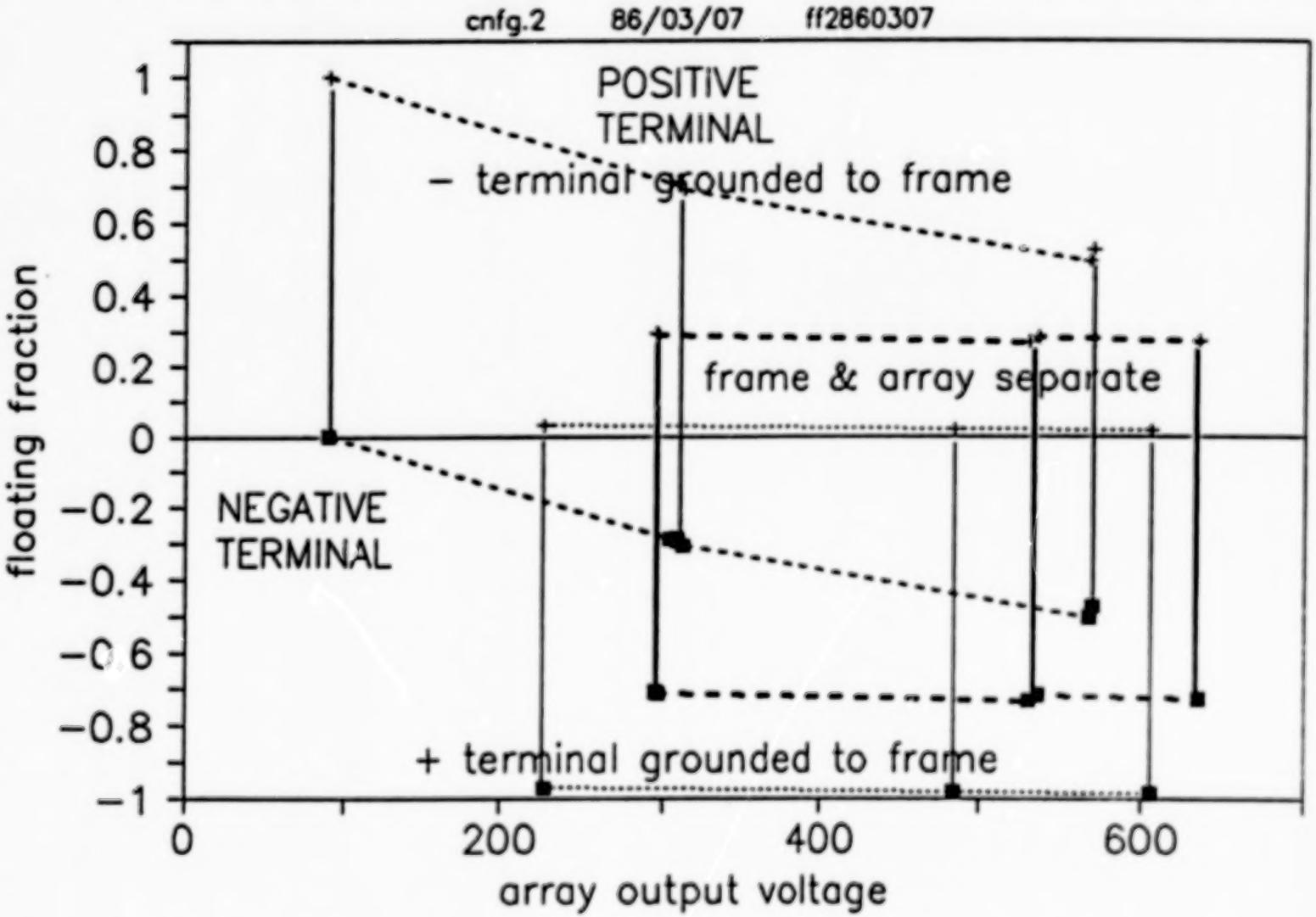
If Positive and Negative ends are widely separated the system floats mostly negative so that:

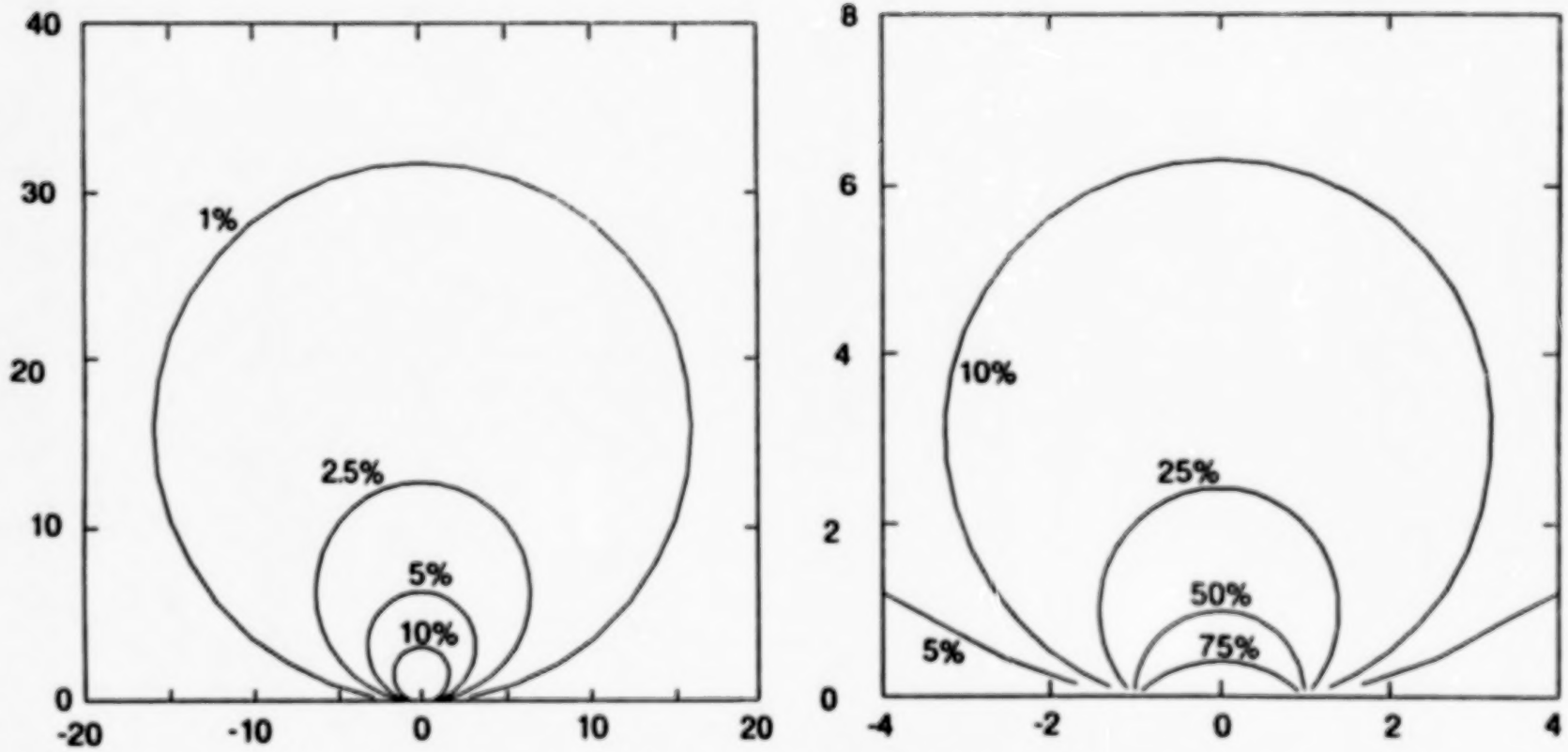
$A_{-\ sheath} * J_{ion} = A_{+\ sheath} * J_{e-}$ (thin sheath approximation)

If Positive and Negative ends are close together compared to thermal sheath thickness, the average potential over the biased area is a few kT negative.

Large Areas of exposed conductor will determine the Floating Potential for the system.

Floating Potentials of Self-Biased Solar Array

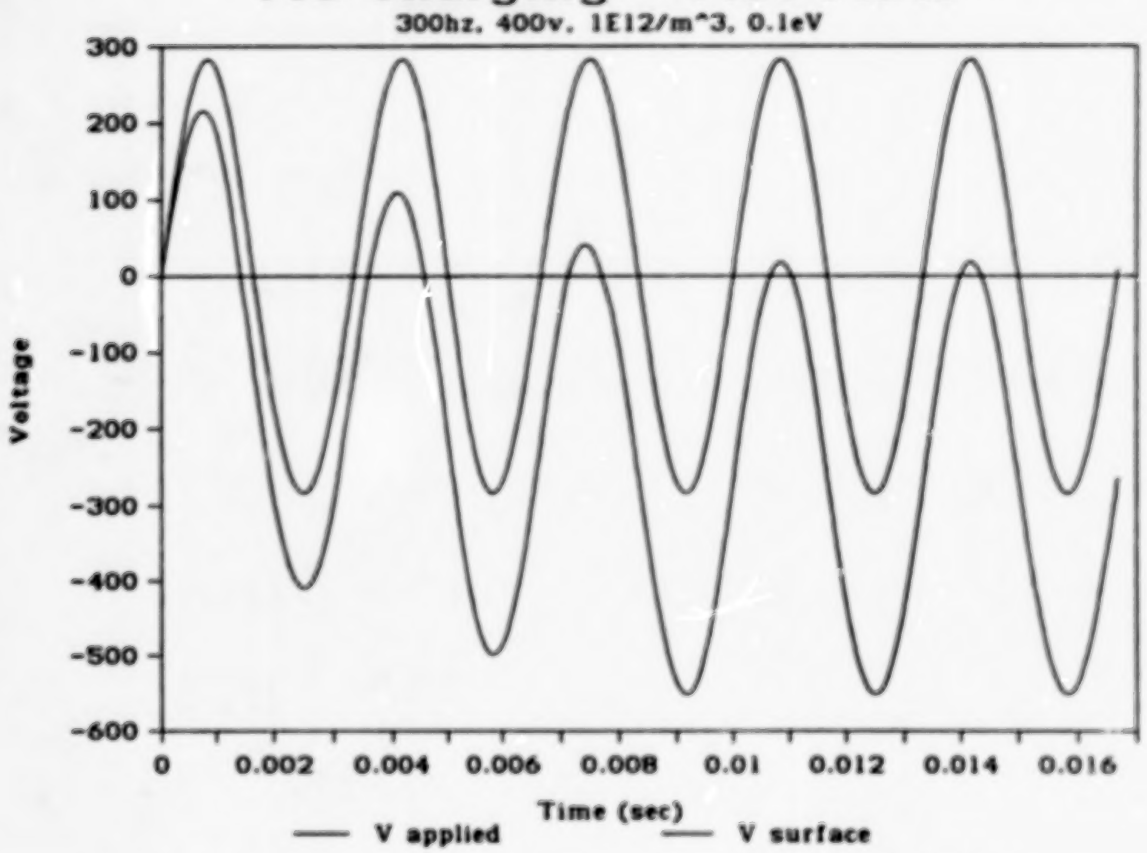


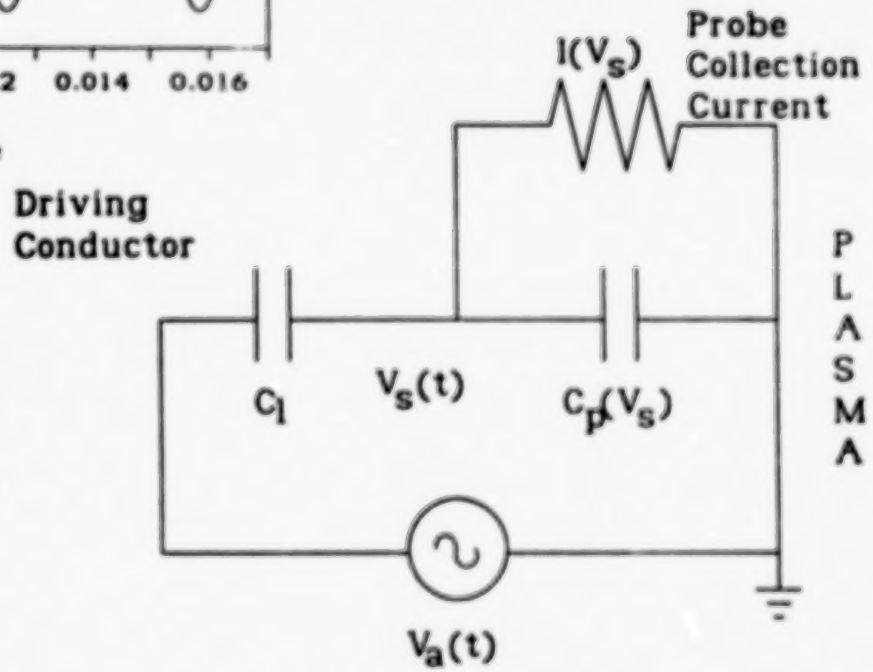


Electric Potential Contours for a biased slit

Solutions to LaPlace's Eqn. (i.e. no Plasma)

AC Charging - Flat Plate





Conclusions

- GEO requirements are not directly applicable to LEO
- Environment Issues for Operating on Large Spacecraft
  - EMI levels
  - v×B·l induced voltage offset
  - Fluctuations of S/C ground relative to plasma
- Environment Issues for external payload design
  - Current Collection from Plasma
  - Discharges
  - EMI radiation
  - Sputtering

PILOT'S ASSOCIATE

G. Edwards
ISX Corporation

(Paper not provided by publication date.)

COOPERATING INTELLIGENT AGENTS

S. Rosenchein
Teleos Research

M. Zweben
NASA/Ames Research Center

(Paper not provided by publication date.)

A TASK CONTROL ARCHITECTURE FOR AUTONOMOUS ROBOTS

Reid Simmons and Tom Mitchell

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We present an architecture for controlling robots that have multiple tasks, operate in dynamic domains, and require a fair degree of autonomy. The architecture is built on several layers of functionality, including a distributed communication layer, a behavior layer for querying sensors, expanding goals, and executing commands, and a task level for managing the temporal aspects of planning and achieving goals, coordinating tasks, allocating resources, monitoring, and recovering from errors. Application to a legged planetary rover and an indoor mobile manipulator is described.

INTRODUCTION

We are currently developing a general-purpose task control architecture (called TCA) for controlling mobile robots. TCA is designed specifically for robots that need to be fairly autonomous and that operate in dynamic and uncertain environments, have multiple goals to achieve, have a variety of strategies to achieve those goals, and use a variety of sensors with different ranges and resolutions.

We are developing TCA concurrently on two testbeds — the CMU six-legged Planetary Rover, and the Hero, a commercially available mobile manipulator platform. The CMU Rover project is an attempt to develop an autonomous robot that can survive, navigate, and acquire samples on the Martian surface [1]. The Hero testbed is an indoor platform that is being used to study coordination of planning, execution, monitoring, error recovery, decision making, and user interaction [3].

TCA is a distributed architecture with centralized control. Communication occurs via coarse-grained message passing between modules, with all messages being routed through the central control. When the central control receives a message, it decides when the message should be attended to, and which module will handle it. We believe that a centralized control scheme will facilitate the coordination of the complex tasks needed for autonomous behavior on Mars. Due to its deliberative nature, however, TCA is not well-suited for robots that need very fast reflexes (e.g., race-car drivers). Our group is beginning research, however, on combining reflexive actions with a deliberative architecture.

TCA can be thought of as a high-level robot operating system — providing a shell for building specific robot control systems. Like any good operating system, the architecture provides communication with other tasks and the outside world, facilities for constructing new behaviors from more primitive ones, and means to control and schedule tasks and to manage computational and physical resources. At the same time, it tries to impose relatively few constraints on the overall control and data flow in any particular system. This should enable researchers to experiment easily with different instantiations of robot control schemes.

TESTBEDS

Our long-term goal is to produce a rover capable of reliable and robust behavior on another planet [1]. Such a system would have to be relatively autonomous, since it will receive infrequent commands from Earth (on the order of every 8 hours) and will have significant communication delay (on the order of 30 minutes).

The tasks envisioned for such a rover include 1) navigating to given sites, 2) acquiring rock and soil samples, 3) surveying for sites of scientific interest, such as regions of sedimentary rock or

underground water, 4) mapping the area traversed, 5) diagnosing system malfunctions, and 6) maintaining communication with Earth.

The rover is designed as a six-legged walking robot. The walker, which will stand around 15 feet high, features orthogonal legs and a split body to enable rear legs to recover past forward legs by passing between the body segments (Figure 1). A prototype leg of the walker has been built and is currently being tested. The leg, along with a laser range-scanner, is mounted on an overhead carriage that is free to roll along a rail (Figure 2). The single-leg testbed "walks" by choosing a footfall location based on elevation maps computed from laser scans, moving the leg to that location while avoiding obstacles, and pulling the carriage with the shoulder and elbow joints.

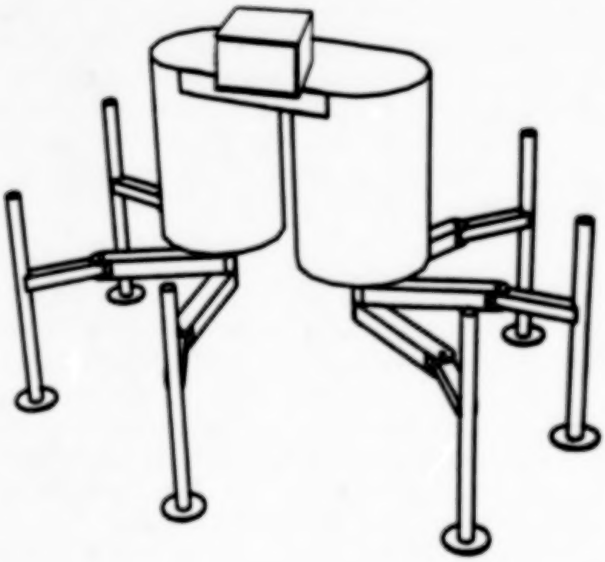


Figure 1. Design of Six-Legged Planetary Rover

Since the current rover testbed is limited in the tasks it can perform, we are exploring issues of coordinating multiple tasks, monitoring, error recovery, and decision making using the Hero testbed [3]. The testbed is based on the Heath/ Zenith Hero 2000, a wheeled robot with manipulator arm and on-board sonars (Figure 3). In addition, an overhead camera provides a two-dimensional plan view of the lab.

Our goal is to let the Hero operate unattended for hours or days at a time in our lab and nearby vicinity. The high level goals of the robot will include 1) collecting cups on the lab floor and placing them in a receptacle, 2) retrieving printer output when requested and delivering it to a workstation, 3) avoiding obstacles, 4) recharging batteries when necessary, and 5) exploring and mapping its environment, when not otherwise occupied. We have already implemented the first three tasks, and are working on the other tasks, as well as the problems of coordinating multiple tasks and error detection and recovery.

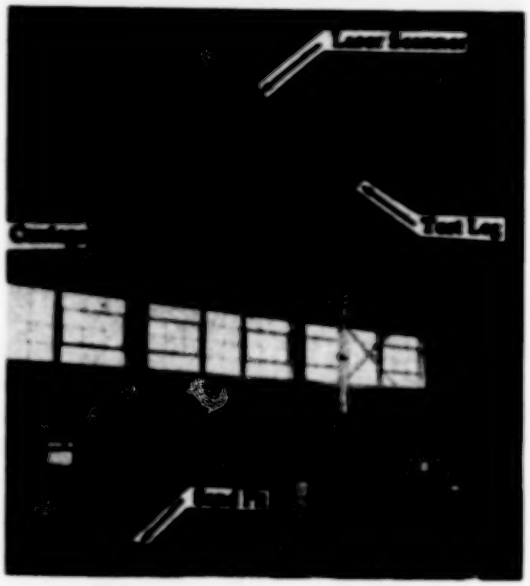


Figure 2. Single-Leg Testbed



Figure 3. Hero 2000 Robot

SCENARIO

This section presents a scenario for the Hero robot that illustrates capabilities we want TCA to support. Similar scenarios are envisioned for the Planetary Rover.

The scenario begins with the Hero using its overhead camera to spot two cup-like objects on the floor. It forms two "cup-collection" goals

and prioritizes them by choosing to attend to the closer object first. The robot plans and executes a path to the object. While moving, it monitors for objects in its path; at the same time, it uses its overhead image of the lab to pre-plan a path from the object to the receptacle. Upon arriving near the object, the robot uses sonar sensors to determine the height and width of the object. If this matches its model of a cup, it plans how to pick up the cup, then executes the actions. The robot then uses the plan previously made to navigate to the receptacle, where it drops off the cup.

Next, the robot attends to collecting the other cup it had spotted previously. While moving towards the second cup, it receives a printer retrieval request. Since this request is of higher priority, the robot suspends planning and execution of the cup-collection goal. It plans and executes a path to the printer, monitors the printer to determine when printing is finished, and then picks up the paper. While waiting for the printer, it plans a path to the appropriate workstation. In this case, the workstation is located outside the lab, so it plans a path to the door, with the intention of using its sonar-based navigation once outside the room.

Before executing the plan, however, the robot notices (from its overhead vision) that an obstacle has appeared in a segment of the path. The robot attempts to replan that segment to detour around the obstacle. If no detour can be found, the robot replans higher level segments until a clear path is found.

At this point, the robot notices that its battery charge is getting low. It estimates that it has enough power left to deliver the output before it needs to recharge, so it continues towards the door. Upon finding the door closed, it re-prioritizes its goals and, after setting the printer output down near the printer, docks with its charger and waits until the battery is charged.

CAPABILITIES

The above scenario illustrates many of the capabilities we believe will be important for autonomous robots, capabilities that we desire the task control architecture to support.

Achieving Goals — the most basic need of a robot is to construct and execute plans to achieve given goals, such as collecting cups. The robot constructs plans based on the current (or projected) environment, available resources, and its other goals and beliefs. It should also be able to execute parts of its plans before specifying them completely, for instance, moving towards a potential cup before determining exact grasp points.

Coordinating Multiple Tasks — for autonomous robots with many goals but limited resources, prioritizing and scheduling its various tasks are crucial. The robot should make decisions about which goals to attend to by comparing their relative costs, benefits, likelihood of success, etc.

Reacting to Environmental Changes — if a new cup is placed on the floor, or the battery charge is low, the robot should notice the change in a timely manner and attend to it, if necessary. Handling such changes may involve adding new subgoals (e.g., collect a new cup), re-prioritizing tasks (e.g., stop collecting the cup and go recharge), or replanning.

Error Recovery — part of reacting to changes includes noticing when a plan or action is failing. The robot should have general mechanisms for recovering from both execution and plan time errors. For example, if an obstacle appears in the robot's path, it might want to detour around the object, or plan a new path to its goal. Similarly, if the robot is unable to find a path, it may try replanning with less restrictive constraints (e.g., allowing more tolerance), or may just attend to a different goal altogether.

External Communication — although not illustrated in the scenario, interaction with humans is a necessity. The robot should be able to explain its decisions and actions. It also should request assistance when needed and allow people to add goals and to alter plans and decisions made by the system.

MECHANISMS

To facilitate experimentation with different control schemes, TCA is built as a layered system. The current layers of functionality include mechanisms for 1) communicating between distributed processes, 2) building behaviors out of more primitive behaviors, and 3) managing the planning and execution of tasks.

TCA is designed so that an implementor can choose which layers to use — higher layers

provide more functionality specific to robot control, but lower layers provide flexibility to implement alternative control schemes. For example, if the implementor finds the types of messages in the behavior layer insufficient, s/he can construct new types using the lower communication layer.

Communication Layer

The base layer of TCA supports the sending and receiving of messages between distributed processes (modules). Modules send messages to a central control module, which routes the messages to the appropriate modules to be handled. TCA supports the use of a variable number of modules — in fact, modules can be added or removed while the system is running. The communication layer contains mechanisms to route messages, notify modules when messages are pending, and send and receive data, even between modules written in different languages (currently both Lisp and C are supported).

Using a user-supplied description of the format of a message, TCA translates the message data into a linear stream of bytes, routes the data, and then reassembles it on the receiving end. All data transfer is transparent to the user. The data format language we developed allows for primitive data types (e.g., integer, float, string), and composite types (e.g., structures, arrays, pointers).

Communication via message passing encourages the use of good software engineering techniques — first defining the functionality and interfaces of the system, implementing them in a modular fashion, and treating the functions as "black boxes." We have found that this eases the effort to integrate different parts of the system. This contrasts with some architectures in which some functions are allowed to interfere with the internal workings of others (e.g., [2]).

Note that while perception, planning, and execution modules are distributed, control of how messages are handled is centralized. A major advantage of centralized control is that it facilitates coordination of the robot's behavior. All control decisions, such as which goals to pursue, or which modules should handle particular messages, are made centrally, where global information can be used to determine the best alternatives.

A potential problem with centralized control is that it may become a bottleneck. This may be overcome through conventions, such as using coarse-grained behaviors to limit the amount of process-to-process communication, and limiting the amount of information passed in each message. Although further experimentation might show that this is indeed a problem, the current message passing cycle time of around 50 milliseconds has proven to be sufficient for our applications.

Behavior Layer

TCA provides several types of primitive building blocks needed to construct robot behaviors. The primitive behaviors are implemented as different classes of messages, built on top of the communication layer. The classes differ mainly in their flow of control. For example, query messages block the user's code until a reply is received, while goal messages are non-blocking and report success or failure directly to the central control.

Query messages are requests to provide information about the external or internal environment, such as computing an elevation map or determining the robot's current position. Query messages are routed either to modules that have access to external and internal sensor data, or to the constraint data base (see below). The module issuing the query suspends execution pending the reply.

Goal messages are intended to support top-down, hierarchical planning. A typical response to a goal message would be to issue other (sub)goal or command messages based on the results of some queries.

Unlike queries, goal messages are non-blocking. That is, the central control may queue the goal until resources become available; in the meanwhile, the module sending the goal message can continue. This implies that a planning behavior cannot assume that the goal will actually be achieved after the goal message is issued. The rationale is that non-blocking goal messages give the implementor greater flexibility in controlling the achievement of goals, such as planning in advance of execution. If goal messages were blocking, planning would always be depth-first — the first subgoal would have to be completely planned before the next subgoal message could be issued.

Command messages, which are used to execute actions, are similar to goal messages. The difference is manifest only at the task layer, which distinguishes between order in which goals are planned and the order in which commands are achieved. For example, although the robot might be able to plan how to go from the printer to a workstation before planning how to pick up the printer output, it obviously should not actually go to the workstation before it has the output in hand.

Constraint messages provide a way to alter the robot's internal state, just as command messages alter the external environment. For example, constraint messages can be used to set the robot's desired average speed, or add expectations about its future behavior. We plan to implement a global data base (blackboard) to facilitate adding constraints and maintaining consistency among them. Currently, constraint messages are used to set global variables, whose values can then be accessed via queries.

Task Layer

The task layer provides mechanisms for maintaining hierarchical goal structures, allocating resources, monitoring the environment, recovering from execution and plan-time errors, and coordinating multiple tasks. The main representations in the task layer are goal structures, resources, and monitors.

While the behavior layer defines goal and command messages, the task layer contains mechanisms for constructing and analyzing goal hierarchies. For each goal, TCA maintains a subtree of the goal, command, and monitor messages (and their descendants) issued by the goal. Facilities exist for tracing goal/subgoal relationships, displaying the goal structure, and suspending or killing subtrees (needed for switching tasks and doing error recovery).

For the scheduling of tasks, TCA contains a general facility for reasoning about time [4] that enables modules to temporally constrain the planning and achievement of goals. A module specifies constraints on the planning intervals of goals (the time needed to completely expand a goal subtree) and the achievement intervals (the time needed to execute all the commands of the subtree). For example, a module might specify that the achievement of G1 precedes the achievement of G2, but that the planning of G2 precedes that of G1. Similarly, it might constrain G3 to be completely planned before any of its sub-commands can start being achieved (by default, planning and execution can occur concurrently). This temporal framework should enable implementors to take advantage of concurrencies in the distributed environment of TCA — for instance, planning routes from a given area while still travelling to the area.

It is crucial for an autonomous agent to effectively allocate its limited resources in order to satisfy its goals. The robot must detect when tasks need competing resources, and must prioritize and schedule tasks when conflicts occur. In TCA, a resource is an abstract entity that is used to manage the handling of messages. A resource may be associated with a computational entity, such as a module, or with a physical entity, such as a motor or range-scanner. Resources are created with a capacity — the number of messages the resource can handle simultaneously. By default, TCA associates a single resource of unit capacity with each module. In addition, a module can create additional resources and associate message handlers with them.

A message received by TCA is queued until the resource that handles the message has available capacity. Currently, messages are handled in FIFO order, subject to the temporal constraints imposed by the goal structure. In the future, we plan to add mechanisms for prioritizing messages. Since the prioritization is context dependent, it will be determined by user-supplied functions, accessed using "decision messages." A module can also explicitly reserve a resource, temporarily preventing other modules from using the resource. While taking an image, for example, a vision module might reserve the "robot motors" resource to ensure that the robot does not move during that period.

Monitors are mechanisms that query for specified changes in the environment, such as obstacles in the robot's path or low battery charge, and take some action based on the results. A monitor is specified by the condition to be monitored (a query message), an action to take if the condition holds (a goal, command, or monitor message), and the time, relative to other messages, when the monitoring is to take place. Point monitors, which test the condition just once, are useful for checking the preconditions or postconditions of an action, such as checking that a planned move succeeded in reaching the desired location. Interval

monitors, which have a temporal extent, are useful for checking for changes in the environment, such as a low battery charge or the appearance of a new cup. Two complementary implementations exist for interval monitors — synchronous polling at a fixed frequency, and asynchronous demon-invocation.

When a monitor detects an error condition, it sends a "failure message" to the central control. The architecture will then decide what to do based on the current environment and the goal structure for the goal that failed. The decision, made by user-defined handlers for the failure messages, may include replanning the goal with additional constraints, replanning a higher level goal, or adding a new subgoal to patch the initial plan. We believe that the goal structure dependencies maintained by TCA will prove useful in diagnosing and recovering from errors [5].

Since monitors must be coordinated by the central control, special reflex behaviors are needed to provide bounded-time reactions to imminent dangers, such as collisions. Such reflexes, which would be implemented outside the centralized TCA, would have a default response (e.g., "halt immediately") and then signal TCA that a reasoned response to the error is required. This strategy is being tested on the Hero testbed. We implemented a "guarded move" routine that checks the robot's encoders and sonars and stops the robot if it detects a collision or impending collision. Once stabilized, the robot notifies TCA so that an appropriate recovery can be planned.

CONCLUSIONS

Currently, we have implemented the communication layer, the behavior layer (except for the constraint data base), and most of the task layer (except for mechanisms to deal with decision and failure messages). We have used the communication and behavior layers of TCA to run the single-leg testbed for the CMU Rover. Our Hero testbed has recently been re-implemented to use the architecture, and we are currently experimenting with it to test out and expand the task layer.

We have described a task control architecture (TCA) that we believe will be useful in implementing control systems for autonomous mobile robots. TCA is designed around a distributed message-passing system that uses centralized control to queue and route messages. TCA is built using layers of functionality to provide flexibility in experimenting with different control regimes. In addition to the message-passing communication layer, higher layers provide mechanisms for building and coordinating complex robot tasks and behaviors, including mechanisms for goal structure manipulation, temporal reasoning, resource management, and monitoring.

ACKNOWLEDGEMENTS

The TCA was designed with the help of the CMU Planetary Rover group, under NASA contract NAGW-1175. In particular, Christopher Fedor and Long-Ji Lin have helped in both the design and implementation of TCA and the Hero testbed. Kevin Ryan implemented the Hero guarded move routine.

REFERENCES

1. Bares, John, Hebert, Martial, Kanade, Takeo, Krotkov, Eric, Mitchell, Tom, Simmons, Reid, Whittaker, William "An Autonomous Rover for Exploring Mars," IEEE Computer, Vol. 22, No. 6, June, 1989, pp. 18-25.

2. Brooks, Rodney, "A Robust Layered Control System for a Mobile Robot," IEEE Journal of Robots and Automation, Vol. RA-2, No. 1, 1986.

3. Lin, Long-Ji, Mitchell, Tom, Simmons, Reid, "A Case Study in Autonomous Robot Behavior," CMU-RI-TR-89-1, Robotics Institute, Carnegie Mellon University, January, 1989.

4. Simmons, Reid, "'Commonsense' Arithmetical Reasoning," Proceedings of AAAI-86, Philadelphia, PA, August, 1986.

5. Simmons, Reid, "A Theory of Debugging Plans and Interpretations," Proceedings of AAAI-88, St. Paul, MN, August, 1988.

AUTOMATED CONTROL OF HIERARCHICAL SYSTEMS USING VALUE-DRIVEN METHODS

Dr. George E. Pugh
Chairman of the Board
Decision-Science Applications, Inc.
1110 North Glebe Road, Suite 400
Arlington, VA 22201-4714
(703) 243-2500

Thomas E. Burke
Director, Command and Control Applications
Decision-Science Applications, Inc.
1110 North Glebe Road, Suite 400
Arlington, VA 22201-4714
(703) 243-2500

ABSTRACT

This paper provides an introduction to the Value-driven methodology, which has been successfully applied to solve a variety of difficult decision, control, and optimization problems. Many real-world decision processes (e.g., those encountered in scheduling, allocation, and command and control) involve a hierarchy of complex planning considerations. For such problems it is virtually impossible to define a fixed set of rules that will operate satisfactorily over the full range of probable contingencies. DSA's value-driven methodology offers a systematic way of automating the intuitive, common-sense approach used by human planners. The inherent responsiveness of value-driven systems to user-controlled priorities makes them particularly suitable for semi-automated applications in which the user must remain in command of the systems operation. Three examples of the practical application of the approach in the automation of hierarchical decision processes are discussed: the TAC Brawler air-to-air combat simulation is a four-level computerized hierarchy; the autonomous underwater vehicle mission planning system is a three-level control system; and the Space Station Freedom electrical power control and scheduling system is designed as a two-level hierarchy. The methodology is compared with rule-based systems and with other more widely-known optimization techniques.

INTRODUCTION

The value-driven methodology described in this paper had its beginnings in the 1960s as an optimization technique for large non-linear, discrete problems; specifically for the allocation of strategic weapons to targets. For example, the assignment of U.S. missiles and bombers to Soviet targets has on the order of 400 weapons types, 30,000 targets, a non-linear objective function, and is defined over a discrete space.

Over the past two decades, it has been successfully applied to an exceptionally wide variety of very complex automation and control problems, including:

- Decision and control systems for autonomous robotic vehicles
- Network design and optimization for telecommunications systems
- Automated and semi-automated control systems for military command, control, and communications applications
- Production scheduling and optimization systems
- Scheduling and control systems for space applications
- Simulations of human decision processes in computerized combat simulations

DSA's value-driven control methodology focuses on the modeling of the system being represented rather than on the optimization technique itself. Furthermore, many problems cannot be represented as a strict optimization problem either because they are too complex or because optimization is not the objective but rather the desire is to have an automated or semi-automated system that performs the same as an *experienced* user would perform given the same set of conditions.

Over the years of research and application of the methodology, DSA's system development efforts have fallen into four major areas:

- pure optimization systems
- scheduling systems
- control systems--centralized, distributed, and hierarchical
- systems that mimic human decision process

This paper provides an introduction to the concepts underlying value-driven control methods and briefly describes three applications in the automated control of hierarchical systems. We also provide a brief comparison with other more widely known approaches for building intelligent systems.

PRINCIPLES OF VALUE-DRIVEN CONTROL

A value-driven control system is one in which the automated decision processes are governed by value-maximizing principles rather than by the "if-then" rules common to expert systems and traditional automation software. Value-driven systems are designed to maximize a system of priorities, and are automatically and intelligently responsive to real-time changes in the problem environment The systematic value-maximizing process which is used in place of a pre-defined set of decision rules operates essentially as follows:

1. It "considers," or searches over, a set of possible decision alternatives;

2. It evaluates each alternative in terms of the currently specified value priorities, and

3. It selects and implements the alternative that yields the maximum value in terms of the currently specified value priorities.

Use of Decision Entities

Decomposition of a complex problem is an essential step in the formulation of a value-driven system.

Indeed, value-driven systems that are responsible for complex decision and control processes are typically composed of a large number of automated "decision entities", each responsible for the functions within a limited subarea. System behavior is then controlled by a system of value functions, one value function per decision entity. A typical value function will have the functional form

$$V = V(a_i, \ldots a_n, x_i, \ldots x_n) \quad (1)$$

where the a_i's are adjustable parameters controlled by the user or by higher levels in a hierarchical system and the x_i's are the state variables of a projected state corresponding to one possible decision alternative.

Each decision entity will have a suitable representation of the decision environment, so that it can formulate feasible decision alternatives and evaluate the desirability of the alternatives in terms of the currently applicable objectives and priorities. A decision entity will select an alternative that maximizes the value function. In particular, the responsible decision entity must be provided with:

- Appropriate policy guidance concerning the current objectives and priorities, and

- Current situation or status information concerning the assigned realm of decision responsibility.

Goal Orientation Through Modifiable Subgoals

The a_i's of (1) are the means by which the set of priorities and objectives that guide each decision entity are adjusted to reflect the goals of the system. Thus, each decision entity is operating in the realm of subgoals that are modified as the current state of the system changes from decision point to decision point.

The formulation of the priority scheme within a value-driven system and the refinement of the values are essential to the correct solution to a problem. The central objective is to ensure that the structure of values will in fact be effective in achieving the real goal for the system.

Responsiveness to Command Priorities

The inherent ability of value-driven systems to respond intelligently to command priorities makes it possible to develop hierarchical control systems in which each level of an automated hierarchy is responsive to the changing objectives and priorities specified at higher levels. A high level entity, by changing the a_i's in the value function of a lower level entity, changes the lower level entity's perception of the relative desirability of different goals or objectives, and thereby influences the behavior of the lower level's selection of courses of action.

The responsiveness of value-driven systems to user-controlled priorities makes such systems particularly suitable for semi-automated applications in which the user must remain in command of the system's operation. Indeed, in those difficult applications in which a reliable and rapid response is required, the value-driven methodology can provide a practical partnership between man and machine. A partnership that allows the man to retain flexible policy control over the system's behavior, while the automated system provides the required rapid response to real-time contingencies.

Application in Hierarchical Control Systems

Two significant properties of value-driven systems that make them particularly suitable for hierarchical applications are: the provision of a means for controlling the flow of information throughout the system, and the ability of each decision entity to continue operation in the absence of specific instructions from higher levels.

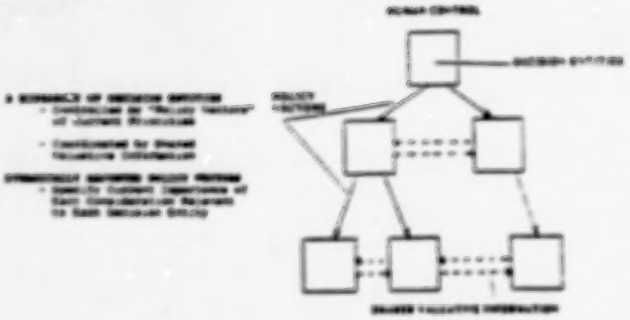


FIGURE 1. Fundamental Concept for Hierarchical Control

The basic design concept for the implementation of value-driven methodology within a computerized hierarchy is illustrated in Figure 1. Each activity is represented by a decision entity, which appears in the figure as a box. Each decision entity is guided by "policy vectors" that are defined at the next higher level in the appropriate control hierarchy. This incoming policy vector defines for the decision entity the current relative importance of the various policy considerations that are to be taken into account in the evaluation of alternatives. In order to coordinate the decisions at any level so as to most efficiently understand the state of the entire system, information is shared among the decision entities at that level, as shown by the dotted lines in the figure.

THE TAC BRAWLER AIR COMBAT SIMULATION

Introduction

DSA 's TAC BRAWLER model is a comprehensive simulation tool which provides a detailed representation of air-to-air combat engagements involving multiple flights of aircraft. Because of the importance of cooperative tactics and the critical role of human factors (such as surprise, confusion, situation awareness, and the ability to innovate tactical responses in unexpected situations) special emphasis has been placed on simulating these aspects of the engagement process.

To date, TAC BRAWLER has successfully reproduced the characteristics of engagements such as the ACEVAL-AIMVAL flight test series, and the manned simulator engagements made in conjunction with the AMRAAM OUE and has helped determine the characteristics that visual display systems for manned simulators must have if they are to provide realistic training in air-to-air combat. Presently, TAC BRAWLER is being extensively used in design studies for the Advanced Tactical Fighter program and for advanced avionics programs.

Modeling Pilot Behavior

The key factor to accurate modeling of air combat is the treatment of the human decision processes which drive the outcome of air-to-air engagements. DSA has developed a dual approach to the modeling of human decision processes which involves both value-driven decision-making and information-oriented decision architecture. This approach provides a practical solution to the problems involved in modeling multiple aircraft combat as in TAC BRAWLER, and includes:

- Explicit Model of Information Flow
  - Sensors
  - Communications
- Realistic Simulation of Decisions
  - Situation Assessment
  - Explicit Mental Model
  - Consideration of Alternatives
  - Based on Judgmental Values

The explicit simulation of the flow of information into and out of each pilot's personal mental model of the situation is key to the successful operation of TAC BRAWLER. Simulation of the pilot's decision process refers to value-driven decision making. Pilots use their mental models to perform situation assessment functions, generate sets of alternative courses of action, and select a particular action for implementation. The choice among alternatives is made on the basis of a judgmental evaluation of the situation that the pilot believes will result if a particular alternative is adopted.

Information-Oriented Architecture

Figure 2 shows the structure of the Information-oriented architecture of TAC BRAWLER. The central status arrays contain the true physical state of the simulation. Each simulated decision-maker has a personal mental status array which mirrors the central status. The imaging, however, is imperfect; a pilot will not know precisely where other aircraft are or exactly how fast they are moving. More important, aircraft and missiles of which he is unaware will be entirely absent from his mental model.

Information arrives in the mental model via sensor events and communications events which simulate visual, radar, missile warning receivers, radio communications, etc. TAC BRAWLER achieves

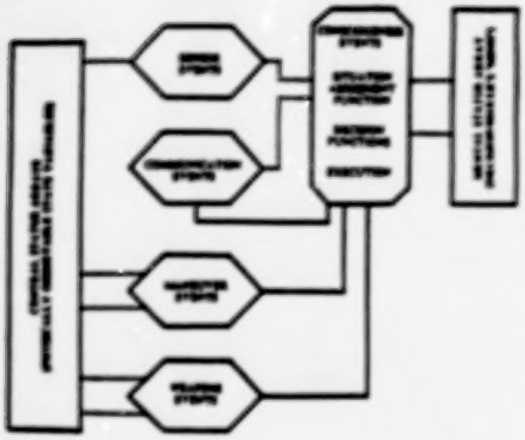


FIGURE 2. TAC BRAWLER Information-Oriented Architecture

a very high level of detail in the modeling of the physical characteristics of aircraft aerodynamics and systems.

Decisions by the pilot cause physical actions to occur either indirectly, via communications, or directly, through aircraft maneuver events and weapons employment events.

Hierarchy of Pilot Decisions

Figure 3 shows the hierarchy of decisions in TAC BRAWLER. The primary effect of high-level decisions is to control the lower level decisions by modifying their evaluation functions and by determining which lower level alternative actions will be considered.

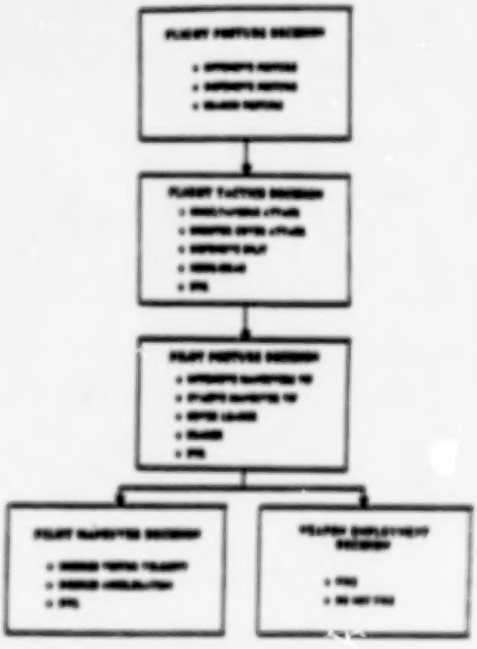


FIGURE 3. TAC BRAWLER Decision Hierarchy

The flight posture decision occurs at the highest level and determines the general course of action. It is made on the basis of a broad description of the situation, such as force ratios and engagement geometry, and also on the basis of user-supplied priorities. At the next level the flight leader determines the tactics that should be used to implement the flight posture. As a result of this decision a specific communication is sent to other flight members, informing them of the tactics. The effect of the message is to influence the values that other pilots use to score the alternative actions they consider. For instance, an order to attack a certain aircraft results in the subordinate perceiving that hostile as being more valuable; he "likes" to attack it. An interesting feature of this value-oriented representation of orders is that pilots can continue to make reasonable decisions in the absence of orders--they have an intrinsic, or default, set of values to use in their decision processes. Additional realism is present, because a pilot's perception of the situation may be very different from that of his flight leader. Since orders only influence a subordinate's mental model and do not force an action upon him, the subordinate exhibits a certain amount of common sense in his actions.

The pilot posture decision determines the general course of action for a pilot: to attack, to evade, to support the flight leader, etc. This decision controls the operation of aircraft systems such as radar and weapons selection, as well as deciding on the current weapons objective.

The maneuver decision and the weapon employment decisions occur at the lowest level. The alternatives considered by the maneuver decision are designed to do things like "get on the hostile's tail", "gain specific energy versus a hostile", "avoid the ground", etc. The weapon employment decision determines whether to fire a weapon at this moment.

Summary

The TAC BRAWLER simulation is the most mature and widely used application of value-driven hierarchical control methods. The model provides a synergy between value-driven logic to simulate pilot decision processes and an information-oriented hierarchical architecture as a means to represent the imperfect knowledge that a decision entity has of the true state of the situation. TAC BRAWLER has become a standard for high fidelity air-to-air engagement modeling within the Air Force, airframe manufacturing, and avionics communities. See reference 1 for a more complete description of the simulation.

INTELLIGENT PLANNER FOR MULTIPLE AUTONOMOUS UNDERWATER VEHICLES

Introduction

DSA has developed a hierarchical value-driven methodology and demonstration prototype system to provide high-level decision and planning functions for the cooperating multiple autonomous underwater vehicles (AUVs) program.

If an AUV (or group of vehicles) is to operate on a truly autonomous basis, then the on-board computer system must be able to accomplish all of the high-level decision functions normally done by the human controller. These decision functions will involve some very complex activities--including even the replanning of the mission itself, in response to unexpected delays or events that may preclude the completion of the mission as originally anticipated. The on-board system must be able to develop and evaluate major revisions in the routing for the mission and it must be able to assess the need to make major changes in the basic mission plan--including even the omission of some or all of the originally planned tasks.

The AUV Operating Environment

The specific decision processes that are needed for the control of an autonomous vehicle depend fundamentally on the environment in which the vehicle is to operate--and on the types of threats and opportunities that are expected. The demonstration includes descriptions of: the natural environment for water depth profile and sonic transmission properties, and; individual vehicle descriptions of estimated position, energy state, and messages awaiting transmission. The postulated test scenarios accounted for potentially hostile objects such as:

- Mines and mine field areas
- Sonobuoys and sonobuoy areas
- Ships or submarines with active or passive sonar

The vehicles are concerned with the planning and execution of missions within this combined environment, in which the specified objectives may include:

- Specified area search and mapping
- Target identification and attack
- Detection avoidance
- The delivery to the operating base or host ship of any significant information from the mission.

Illustrative Tactical Options

The character of the decision-making functions is also dependent on the types of tactical options that are available to the vehicles. Individual vehicles can travel at low speeds (low noise), skirt areas of possible sonar detection, and hide in the bottom clutter. When travelling as a group in dangerous areas, these single vehicle tactics can be augmented by cooperative multivehicle tactics; for example:

- A "high low" tactic
- A "leader-follower"
- A "leap frog" tactic
- A separation with planned rendezvous.

Hierarchical Planning and Control Architecture

For the AUV project, DSA developed three levels of control which closely parallel the normal division of responsibility that is typical of a human command hierarchy: Mission Level, Group Level, and Vehicle Level.

MISSION LEVEL--THE GROUP PLANNERS

The mission level of the system receives a mission definition and basic values from the human Mission Planner. The tasks comprising the mission consist of such items as: search and destroy, reconnaissance, or map. These high level commands will then be broken into a series of subtasks to be performed by the group(s). It is the job of each Group Planner (at the mission level) to devise the actual group tactics and priorities to be employed in carrying out each of the mission subtasks.

GROUP LEVEL--THE VEHICLE PLANNERS

The Group Level decision entity takes the instructions from the Mission Level and turns them into detailed instructions for the Vehicle Levels below. There is one Vehicle Planner for each vehicle in the group, and an overall Planner Manager for the group as a whole. The Planner Manager will make decisions such as which vehicle is the leader and which is the follower in a "leader follower" tactic. A Planner Manager will also plan a route from the current position to the destination specified by the Mission Level.

The Group Level sends instructions to the Vehicle Level in the form of: a desired final location, a desired velocity, a desired depth, and a desired tactic.

VEHICLE LEVEL

The Vehicle Level decision entity takes the instruction from the Group Level and routes the vehicle along a detailed route at the chosen speed and depth. It is, in effect, the navigator of the system, plotting a course to the chosen destination (within the range of its sensors), moving the vehicle around small obstacles detected by the sensors.

The Vehicle Level decides on a optimal route for the vehicle based on its knowledge of the terrain, either through its preset information data base or through knowledge gained by its sensors.

Summary

DSA's prototype demonstration system implements the concepts described above, and shows how at each level the appropriate Planners make decisions based on a trade-off among the many factors that must be considered in mission planning, such as: urgency of mission, risk of detection, and safety of the vehicle.

SPACE STATION ELECTRIC POWER SCHEDULER

Introduction

DSA has recently completed a design concept for the automated control and scheduling of the Space Station Freedom (SSF) electric power system. The concept is different than that of the previous two examples in that it postulates the entire SSF environment as a free market economy where buyers and sellers of resources must bargain for their best options.

The electric power control system is only one of many automated subsystems that must be coordinated to provide a productive environment aboard SSF. To dispatch electric power to satisfy the demands of users without violating any resource constraints will require cooperative problem-solving among the subsystems, payloads, and the OMS to maximize productivity.

Planning and Scheduling

Reference 3 contains a detailed mathematical description of the value-driven approach. The approach may be described physically as a planning hierarchy of automated agents operating in a free-market environment:

- Computerized Resource Management Agents, one for *each* of the managers who are responsible for supplying resources such as electric power, thermal control, or life support for the Space Station.

- Computerized Resource Requesting Agents, one for *each* of the projects or activities that are major consumers of Space Station resources.

- A Free-Market Coordinating Agent responsible for managing and expediting the operation of the free-market as a resource allocation and scheduling mechanism.

The market coordinating agent initiates the process by postulating a trial set of time-dependent prices for each major resource. He then polls the negotiating agents for their individual responses. Each independent agent then responds with a specific plan for his own area of responsibility that would provide the best results for his area, given the postulated price structure.

The resource consuming agents respond first, by specifying how they would schedule their activities to maximize the profitability of their activities on the basis of the postulated price structure. Naturally, insofar as their benefits are not sensitive to the schedule they will schedule their activities to minimize their resource costs. But where the timing of an activity is important to its projected benefit or probability of success, the agent will select a scheduling alternative that maximizes the "profit" (i.e. the excess of the research benefit minus the resource costs). The combined scheduling responses of all of the resource consuming agents defines the overall schedule of consumer demand for each resource that would result from the postulated price structure.

The resource supplying agents respond next. Each agent responds with a specific plan for the delivery of his own resource (here, electric power) that would be most appropriate in the context of the foregoing schedule of consumer demand and the trial prices. If it is feasible to meet the schedule of demand, the resource supplying agent seeks to do so in the best and most efficient way, and he provides an estimate of the fair price that he would have to charge per unit of resource in each time period, in order to meet the specified demand.

Based on these combined responses, the market coordinator assesses the supply-demand and pricing relationships for each resource in each time period and adjusts his trial prices accordingly, either raising or lowering to bring supply in line with demand. He then again polls the agents for their responses.

The free-market coordination process is repeated until a price structure is found which shows a satisfactory relationship between the supply and demand for all resources in all time periods. When con-

vergence is achieved, the free-market coordinator's trial prices will be equal (within an acceptable error tolerance) to the prices as estimated by the individual resource management agents. Therefore, in the final coordinated schedules the supply-demand relationships for each resource will properly reflect the resource supplier's expert estimate of both the costs and the risks of meeting an additional increment of demand. The estimation of these operating costs and the operating risks for various levels of demand (taking into account appropriate margins of safety) is, of course, one of the major responsibilities of the technical managers for the major Space Station resources.

Hierarchical Structure

In this approach the priorities, or values, of each project are defined in terms of a mathematical function whose parameters are divided into two major classes--those that must be controlled by the overall program manager, and those that can be controlled independently by the technical managers for the individual Space Station projects. The two classes of value parameters include the following types of considerations.

1. Parameters Controlled by the Overall Program Manager

   - The relative importance or priority assigned to each project

   - The estimated time urgency of each project

2. Parameters Controlled by the Technical Manager for Each Project

   - The relative technical value or utility of alternative research opportunities involving differences in the duration of the time allotted, in the specific orbits in which the time is allotted, in the specific time allotted within an orbit, and in the allotted envelopes of power and thermal consumption, etc.

   - The dependence of the technical value on other factors associated with scheduling dynamics, such as the time interval between allotted research opportunities, the cost of departing from a previously defined schedule, the importance of not interrupting a research activity once it is started, and so on.

The use of a formalized value structure allows the overall program manager to influence the relative scheduling priorities, without distorting the detailed scheduling preferences of the individual project managers; and it allows individual project managers to specify as broad or as narrow a range of scheduling preferences as they deem appropriate for their project. Perhaps most importantly, the approach provides an appropriate dynamic response to variations in the resource prices in which an increase in price can remove the low-priority or low-urgency projects from the scheduling competition, while at the same time allowing the most urgent or highest priority projects to continue at their required operating levels.

COMPARISON WITH EXPERT SYSTEMS

In its domain of application, value-driven control methods overlap the domain of the branch of Artificial Intelligence known as "expert systems." Expert systems focus on the actions to take at each step of a process. A system of rules is developed (often called situation-action rules) that determine what action to take, perhaps to consider another rule, at each step.

In contrast to this focus on the steps in the process, value-driven systems focus on the overall objective of the process. That is, value systems are goal-oriented, which has several advantages. For many systems it is desirable to be able to compare different policies. For example, in studying pilot behavior it is important to compare aggressive strategies--with a consequent increase in hostile kills--to more conservative strategies--with a consequent increase in friendly survivors. Or, more generally, it is often extremely important for a user to know what objective is being optimized. In rule-based systems this is often difficult to discern, particularly when new rules are added to a system.

Complex systems are frequently easier to model using value-driven techniques. For example, the Air Force model that was in use prior to TAC BRAWLER was rule-based; but it modeled only two aircraft. That is, it was a one-on-one model. Generalizing that model to represent multiple aircraft proved totally unwieldy; there were simply too many rules that had to be developed. The need for an exhaustive and consistent set of rules completely overwhelmed the situation. A value-driven system whose objectives are to a large extent independent of the number of aircraft proved much more efficient.

On the other hand, there are many situations that lend themselves very naturally to being represented by rule-based systems. These tend to have a linear or sequential nature. The medical diagnosis system MYCIN is of this type. For this type of system it makes a great deal of sense to proceed through a series of steps to arrive at a correct diagnosis, rather than attempting to optimize some difficult to define value function.

The significant difference between value-driven systems and expert systems seems to lie in the linear and sequential nature of the problems that lend themselves to efficient treatment by rule-based techniques. For, by contrast, problems that lend themselves to treatment by goal-oriented techniques tend to have a collective nature to them; no single (simple) path can be defined that leads directly to the solution.

COMPARISON WITH MATHEMATICAL PROGRAMMING

Value-driven theory is an outgrowth of a very general method for optimizing large systems that are characterized by non-linear objective functions, are defined over a discrete space, and are subject to inequality constraints. The method known as Generalized Lagrange Multiplier (GLM) Theory, and described in reference 4, is a generalization of the classical Lagrange method. As such, value-driven systems are a form of mathematical programming. However, value-driven systems, including even those that are specifically designed as optimizing systems, are typically quite different from the more widely known techniques in both their design philosophy and their operating characteristics.

In the application of most optimizing methods, it is usually necessary to make simplifying assumptions in the problem structure, to recast it in a standardized form that is compatible with an established procedure. That is, it is standard practice to sacrifice accuracy in the representation of the problem, in order to provide a rigorous and accurate mathematical optimization.

In a typical value-driven system, the design priorities are reversed. That is, the achievement of a precise mathematical optimum is rarely a driving objective in the problem formulation. Rigorous optimization tends to be sacrificed when necessary, to provide a more accurate representation of the problem including:

- A more accurate representation of the physical problem and its associated constraints, and

- A satisfactory representation of all of the valuative considerations that realistically must be taken into account in order to select an appropriate real-world course of action.

The focus on problem representation is aided immensely by some features of the GLM method itself. The lack of restriction on the objective (value) function--it can be almost any function imaginable--gives the developer great flexibility in realistically representing the complex objectives that must be reflected in real-world systems.

SUMMARY

The distinctive features of the value-driven methodology: focus on system-wide goals, responsiveness to command priorities, decomposition to decision entities, and applicability to large-scale systems, makes it possible to develop automated systems that can cope effectively with complex planning and control tasks that can be addressed only through a hierarchical decision process.

The three program described illustrate the three stages of system development for hierarchical control systems. A conceptual design for the automated scheduling of the Space Station Freedom electric power system. A prototype demonstration system for cooperating multiple autonomous underwater vehicles. A mature widely used simulation of multiple aircraft air-to-air combat.

REFERENCES

1. Kerchner, R. M., "The TAC BRAWLER Air Combat Simulation Management Summary Manual (Rev 5.0)," DSA Report #907 for HQ USAFCSA/SAGF, Contract F49642-87-C-0022, Decision-Science Applications, Inc. Arlington, VA, June 1988.

2. Krupp, Joseph C., "Providing Intelligent Planning for Multiple Autonomous Underwater Vehicles," DSA Report # 839 for National Bureau of Standards, Contract 50SBN6C4130, Decision-Science Applications, Inc. Arlington, VA, September 1987.

3. Pugh, George E., "A Value-Driven Electric Power Scheduler," DSA Report #977 for NASA Lewis Research Center, Decision-Science Applications, Inc. Arlington, VA, June 1989.

4. Evert, Hugh III, "Generalized Lagrange Multiplier Method for Solving Problems of Optimal Allocation of Resources," Journal of the Operations Research Society of America, Volume III, No. 3, May-June 1963.

Planning for Execution Monitoring on a Planetary Rover

Erann Gat R. James Firby David P. Miller

Jet Propulsion Laboratory/California Institute of Technology
M.S. 301-440
4800 Oak Grove Drive
Pasadena, California 91109
USA

Abstract

A planetary rover will be traversing largely unknown and often unknowable terrain. In addition to geometric obstacles such as cliffs, rocks, and holes, it may also have to deal with non-geometric hazards such as soft soil and surface breakthroughs which often cannot be detected until rover is in imminent danger. Therefore, the rover must monitor its progress throughout a traverse, making sure to stay on course and to detect and act on any previously unseen hazards. Its onboard planning system must decide what sensors to monitor, what landmarks to take position readings from, and what actions to take if something should go wrong. This paper describes the planning systems being developed for the Pathfinder Planetary Rover to perform these execution monitoring tasks. This system includes a network of planners to perform path planning, expectation generation, path analysis, sensor and reaction selection, and resource allocation.

1. Introduction

Efforts are currently underway to develop an autonomous mobile robot for the unmanned exploration of planetary surfaces. Such a robot must be able to plan its actions based on sensor data which is inexact and incomplete. Furthermore, there are non-geometric hazards such as dust pits and unstable slopes which cannot be detected reliably with remote sensors. Therefore the robot must possess a robust execution monitoring system which will allow it to detect and recover from unexpected occurrences in real time during path execution. The execution monitoring system described in this paper consists of an integrated architecture that includes a number of different planning systems working together.

There are several issues which must be addressed when designing an execution monitoring system. First, the computational resources available at run time may not be sufficient to constantly monitor all of the vehicle sensors at once. Therefore the system must choose judiciously which sensors to monitor and with what duty cycle to monitor them. The system also must schedule the operation of sensors such as cameras or rangefinders which may require significant amounts of time for aiming and data processing. Ideally, when an unexpected sensor reading is encountered, the system should be able to diagnose the source of the problem and take appropriate corrective action. This must occur in real time as sensor violations could indicate that the vehicle is in imminent danger. The rover must not compute for an hour to decide to back out of a dust pit into which it is sinking. Finally, the use of shared resources during execution monitoring must be coordinated with the other subsystems that use those resources (*e.g.*, cameras might be used by the science subsystem as well as for navigation).

This paper describes an execution monitoring system currently under development which addresses many of these issues. The system is integrated into an autonomous path-planning and execution system which controls a six-wheeled vehicle traversing rough outdoor terrain. Section 2 gives an overview of the entire vehicle control system. Section 3 describes the execution monitoring runtime system which monitors the vehicle during a path traverse. Section 4 describes the execution monitoring planner which produces the execution monitoring profiles that control the runtime system. Section 5 presents an example. Section 6 summarizes.

2. System Overview

In the semiautonomous navigation (SAN) approach which we are investigating, local paths (five to ten meters in length) are planned autonomously using local sensor data obtained by the vehicle. This local path planning is guided by a global route which is planned off-line using a low-resolution topographic map. The global route takes the form of a potential field defined over a region between

the rover's starting location and goal [Payton88, Arkin89]. After the local path is generated, it is simulated to generate sensor expectations and appropriate reflexes are set up for execution when a sensor expectation is violated. Finally, the path plan, including expectations and reflexes is made available for execution. The various steps in this process are coordinated by a system executive. A block diagram of the overall system is shown in figure 1.

The system operates in cycles. At the beginning of a cycle, the system executive instructs the vehicle's sensing and perception system to construct a model of the terrain surrounding the vehicle. This model is based on information from stereo cameras, laser rangefinders, and a low-resolution database provided by an orbiting spacecraft. The final local model includes height, slope and roughness information at varying resolutions, and is in a form that is independent of the particular physical sensors used to collect the data [Gennery77, Gennery80, Wilcox87].

The local terrain model is passed to the path planning subsystem along with a goal location from the system executive. The path planner constructs a local path between five and ten meters in length [Miller87, Slack87]. This path is passed to a vehicle simulator which performs a detailed kinematic simulation of the vehicle traversing the planned path. This simulation serves two functions. First, the resulting information can be used by the planner to perform local optimization of the path. This is done by making small changes to the original path and sending it to the simulator again to determine if a more efficient path results [Thorpe84]. Energy to power the rover's motors and computers is a scarce resource so the local optimization continues as long as the energy saved by optimizing the path is more than the energy required to compute the optimizations [Miller89].

The simulator's second function is to produce expected values for all of the physical sensors on the vehicle as it traverses the path. These expected values are used by the execution monitoring planner to construct execution monitoring profiles. These profiles tell the run-time execution monitoring system which sensors to monitor and when to monitor them.

The execution monitoring planner also contains a predictive monitoring system which attempts to identify specific problems which may arise during path execution. When it identifies a potential problem, it inserts a set of monitoring parameters and recovery procedures to detect and deal with the problem should it arise. For example, large areas devoid of rocks may be dust pits. If the rover is about to traverse such an area, the predictive monitor may insert specialized sensor operations into the plan to look for deep dust in that area of the traverse [Linden87, Doyle89].

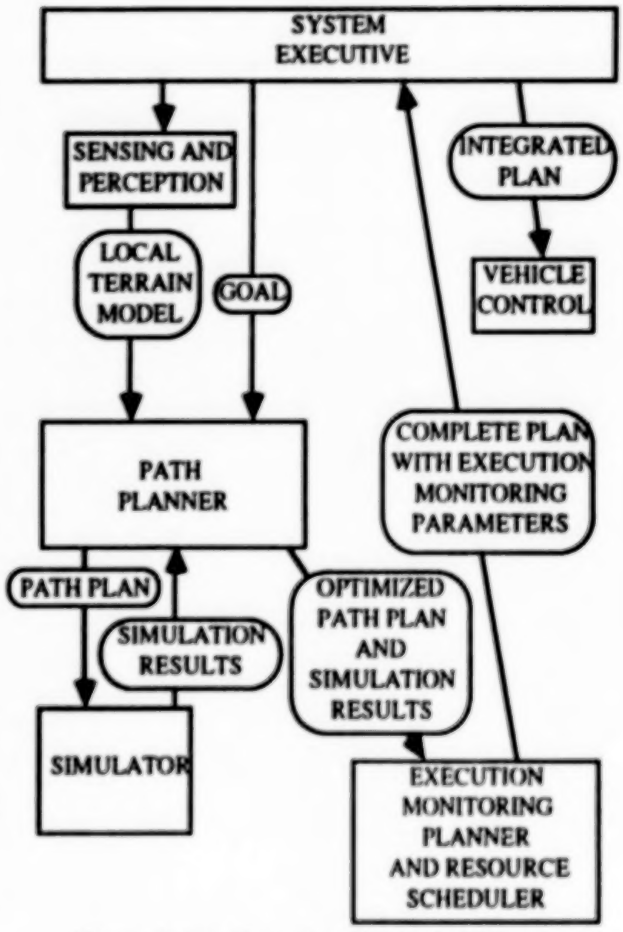


Figure 1: The Execution Monitoring System

The planned path, execution monitoring parameters, and recovery procedures are integrated by the execution monitoring planner into a locally consistent plan using a simple resource scheduler and the result is passed back to the system executive. The system executive checks that this plan conforms to any global constraints that the rover has, such as power limits, shared resource constraints or temporal deadlines. If the plan is acceptable, the system executive passes that plan to the vehicle control system to actually move the vehicle along the planned path. During the traverse, if a sensor reading falls outside of its profile (*i.e.*, an expectation is violated), the vehicle immediately aborts execution of the remainder of the path and executes the recovery procedure associated with that violation (if any).

The system executive then begins a new cycle with the construction of a fresh local terrain model. This may be done during the traverse of a previous path in order to allow interleaved operation of the various subsystems and continuous movement of the vehicle.

3 The Execution Monitoring Runtime System

Vehicle sensors come in three varieties. First, there are physical sensors which do not require resource schedul-

ing, such as wheel encoders and inclinometers. Their values are available continuously to any subsystem which needs them. Second, there are physical sensors which require resource scheduling such as cameras which must be aimed in the right direction at the right time and which require significant processing before useful information is available from them. Finally, there are virtual sensors which are mathematical functions defined over the values of the physical sensors. For example, there are virtual sensors for the vehicle's absolute spatial location in Cartesian coordinates. These values do not correspond to any physical sensor, but are computed using the values of many different sensors. A virtual sensor may require resource scheduling.

From the point of view of the execution monitoring runtime system, no distinction is made between a physical sensor and a virtual sensor. Complex interactions among physical sensors are monitored by setting bounds on a single, specially coded virtual sensor function. Virtual sensors allow the runtime system to be simple and efficient which is essential to achieve real-time performance.

The behavior of the runtime system is defined by a set of execution monitoring profiles computed by the execution monitoring planner. An execution monitoring profile defines an envelope of acceptable values for one sensor, called the dependent sensor, as a function of another, the independent sensor. The envelope is defined by a set of ranges and associated minimum and maximum values for the dependent sensor. The minimum and maximum values specify the limits on the dependent sensor whenever the value of the independent sensor falls in the associated range.

Assigned to each minimum and maximum value is a reflex action to be performed if the value of the dependent sensor should violate one of its limits. The reflex action is simply an index into a table of precomputed reflex actions which can be augmented by the execution monitoring planner. Thus, at runtime, the invocation of a reflex action once a sensor violation is detected can be virtually instantaneous.

By far the most common reflex action is simply to stop the vehicle. However, there are times when this is not appropriate. For example, if the front wheels suddenly start spinning free, and the suspension encoders indicate that those wheels have suddenly dropped, then the front of the rover has probably broken through the surface. If this was not expected, the rover should immediately stop and backup to avoid getting completely mired.

The runtime system can also be used to accurately position the vehicle relative to certain physical landmarks. Suppose the rover needs to position itself one meter from a certain rock in order to collect a sample. This can be accomplished by aiming the rover's rangefinder at the rock and setting up a reflex action to stop the vehicle when the range is one meter. Positioning accuracy can often be significantly improved over simple dead reckoning using such techniques.

4 The Execution Monitoring Planner

The execution monitoring planner uses the local terrain model and information generated by the traverse simulator to produce a set of execution monitoring profiles. These profiles define acceptable ranges for the values of vehicle sensors during the traverse. Whenever the value of a vehicle sensor goes out of the bounds specified by an execution monitoring profile the vehicle immediately executes the reflex action associated with that profile.

The traverse simulator uses the local terrain data, and its uncertainty, to produce expected value ranges for all of the vehicle's physical non-scheduled sensors at points every few centimeters along the path. These values are analyzed by the execution monitoring planner in order to construct a first set of execution monitoring parameters. The planner selects segments of the path where the expected sensor values are more or less constant and sets the limits on that sensor to a value close to the expected deviations predicted by the simulator. The planner attempts to achieve maximum sensor coverage with a minimum of execution monitoring parameters since the performance of the runtime system becomes impaired as the number of parameters grows large.

This initial set of parameters is almost certain to detect a deviation from expected behavior should one occur. However, at runtime, it is very difficult to quickly determine the cause of a problem and decide on an appropriate reflex action using raw physical sensor data alone. Thus, the execution monitoring planner includes a second level of processing to examine the local terrain model and attempt to predict potential problems in the plan. This predictive monitoring system uses a rule-based model of the domain physics which includes information about the likely locations of dust bowls, loose gravel, and other non-geometric hazards. Once the system has identified a potential problem, it finds (or constructs) a virtual sensor, or a set of virtual sensors, to detect that problem specifically and assigns reflexes to handle the problem should it occur.

The predictive monitor also examines the local terrain model for geometric features that it can use as landmarks if special positioning accuracy is required during a traverse. When such landmarks are used, the system generates an execution monitoring profile to check the landmark at strategic points in the traverse, taking into account such things as visibility of the landmark and possibility of confusion with similar nearby landmarks [Chatila85].

All of the execution monitoring parameters generated by these mechanisms are passed to a simple resource scheduler which removes temporal conflicts among shared re-

sources. For example, if many landmarks are to be monitored the traverse plan may have to include delays to allow sensors to be pointed, or there might be more subtle conflicts involving power usage, setup or computation time. In addition, the resource scheduler takes into account some constraints which it may be given by the system executive (*e.g.*, power or time limitations) [Miller86].

Finally, the path description, annotated with the self-consistent execution monitoring profiles, is passed back to the system executive, which then passes it on to the vehicle control subsystem for execution.

5 Example

As an example of the operation of the execution monitoring system, consider the situation depicted in figure 2. The rover path planning subsystem has planned a 10 meter long path that goes between a large rock outcropping to the left of the vehicle and a group of four boulders to the right. The traverse route is mostly flat, with a large open area around the second half of the path. This is passed to the vehicle simulator which generates expected values for the vehicle sensors along the path.

For simplicity we consider only five vehicle sensors in this example, an odometer, an inclinometer, a compass, an elapsed-time clock, and a pointable range finder. From the expected values generated by the simulator, the following execution monitoring parameters could be derived:

Dependent Sensor	Independent Sensor	Range	Min	Max
Inclinometer	Odometer	0 m	-10^{0}	10^{0}
Compass	Odometer	0 m	-45^{0}	10^{0}
Compass	Odometer	2 m	-50^{0}	-40^{0}
Compass	Odometer	4 m	-50^{0}	10^{0}
Compass	Odometer	6 m	-20^{0}	20^{0}
Odometer	Clock	30 sec	9 m	11 m

The first parameter checks the vehicle tilt along the entire path. Since the entire traverse area is fairly flat, all of the inclinometer monitoring is accomplished by a single parameter.

Monitoring the vehicle heading is somewhat more complex. The path is segmented into four pieces. Between 0 and 2 meters the vehicle is turning towards the southeast and so the acceptable range for the heading is quite large. Between 2 and 4 meters the rover travels in more or less a straight line, and so the acceptable range is narrower. There is another transition segment between 4 and 6 meters, and another straight segment between 6 and 10 meters.



Figure 2: A 10 meter path.

The final execution monitoring parameter states that the path must be nearing completion before 30 seconds have elapsed. On the actual vehicle there would be wheel slip sensors which could detect lack of progress long before the end of the path.

These parameters represent the simplest sort of analysis that can be performed on the simulation data: the sensor values are simply analyzed for segments where the values all fall within a certain range. This sort of analysis works well when sensor values are constants, but often creates transition regions where sensor values are not closely monitored, such as the path segments where the vehicle heading is changing. In these cases, the execution monitoring planner could construct a virtual sensor which compared, say, the vehicle heading to the odometer reading (normalized to the start of the transition region) and set up an execution monitoring parameter which monitored the ratio of these two values. Similar correlations can allow nearly every sort of sensor value transition to be monitored as closely as necessary.

Finally, the predictive monitor could insert a number of execution monitoring parameters in this situation. It might, for example, schedule a range reading off the rock outcropping on the left of the vehicle just before the vehicle entered the area between the rocks. This would ensure that the vehicle was not in danger of colliding with a rock as a result of dead reckoning errors. The system might also notice that the large open area towards the end of the path could be a dust bowl, and insert more checks on vehicle articulation. The operation of the predictive monitor is highly heuristic and is based strongly on domain-dependent issues which will be the subject of future research.

6 Summary

An autonomous planetary rover needs a robust execution monitoring system to detect and recover from unexpected occurrences in real time. A system which addresses these goals is currently under development and the Jet Propulsion Laboratory.

The system has two major components, an execution monitoring planner and a runtime system. The runtime system is very simple, which allows it to respond to situations in real time. All of the complex computations are done by the execution monitoring planner before execution begins.

The execution monitoring planner produces execution monitoring profiles which describe acceptable limits on the values of the vehicle's sensors at various stages during the traverse. Vehicle sensors may be actual physical sensors, or they may be virtual sensors which are simply mathematical functions defined over the values of the physical sensors. This allows complex aspects of the vehicle's performance to be monitored efficiently.

The execution monitoring planner derives profiles from two sources. The first is a vehicle traverse simulator which computes the expected values and variances for all of the vehicle's physical sensors at a series of points throughout the traverse. The second is a predictive monitoring system which anticipates potential problems and inserts explicit checks and recovery procedures for those problems.

All of the execution monitoring parameters are passed through a task scheduler to remove conflicts among shared resources. The final, self-consistent traverse plan is sent to the rover's system executive which fits the plan into the vehicle's global plan. If the plan is acceptable, it is sent to the vehicle control subsystem for execution.

During execution the runtime system checks the values of the vehicle sensors against the limits imposed by the execution monitoring profiles. If these limits are violated, the remainder of the path traverse is aborted and a reflex action associated with the violated profile is executed.

Acknowledgements

The research described in this paper was carried out by the Jet Propulsion Laboratory — California Institute of Technology under a contract with the National Aeronautics and Space Administration.

References

[Arkin89] Arkin, R., C., Three-Dimensional Motor Schema Based Navigation, in the *Proceedings of the NASA Conference on Space Telerobotics*, Pasadena, California, Jan31-Feb2, 1989.

[Chatila85] Chatila, R., Position referencing and consistent world modeling for mobile robots, in *Proceedings of the International Conference on Robotics and Automation*, IEEE, pp. 138-145, 1985.

[Doyle89] Doyle, R., S.Sellers and D.Atkinson, Enhancing Aerospace Systems Autonomy Through Predictive Monitoring, *27th Aerospace Sciences Meeting*, AIAA, paper #89-0107, Reno, NV, January 1989.

[Gennery77] Gennery, D.B., A Stereo Vision System for an Autonomous Vehicle, *Fifth International Joint Conference on Artificial Intelligence*, MIT, 576-582, 1977.

[Gennery80] Gennery, D.B., *Modelling the Environment of an Exploring Vehicle by Means of Stereo Vision*, AIM-339, Computer Science Dept. Report STAN-CS-80-805, Stanford University, Ph.D. dissertation, 1980.

[Linden87] Linden, T. & Glicksman, J., Contingency Planning for an Autonomous Land vehicle, in *Proceeding of the 10th IJCAI, AAAI, IJCAI*, Milano, Italy, pp 1047-1054, 1987.

[Miller86] Miller, D. P., Scheduling robot sensors for multisensory tasks, *Proceedings of the 1986 Robots West Conference*, SME, Long Beach, CA., 1986.

[Miller87] Miller, D., P., Slack, M., G., Efficient Navigation Through Dynamic Domains, in the *Proceedings of Workshop on Spatial Representation and Multi-Sensor Fusion*, St. Charles Illinois, October 1987.

[Miller89] Miller,D.P., A.Mishkin, K.Lambert, D.Bickler, and D.Bernard, Autonomous Navigation and Mobility for a Planetary Rover, *27th Aerospace Sciences Meeting*, AIAA, paper #89-0859, Reno, NV, January 1989.

[Payton88] Payton, D., W., Internalized Plans: A Representation for Action Resources, in the *Proceedings of the Workshop on Representation and Learning in an Autonomous Agent*, Lagos, Portugal, November, 1988.

[Slack87] Slack, M. G., Miller D. P., Path Planning Through Time and Space in Dynamic Domains, in *Proceeding of the 10th IJCAI, AAAI, IJCAI*, Milano, Italy, pp 1067-1070, 1987.

[Thorpe84] Thorpe, C. E., Path Relaxation: Path Planning for a Mobile Robot, in *Proceedings of AAAI*, AAAI, pp. 318-321, 1984.

[Wilcox87] Wilcox, W.H., D.B. Gennery, A.H. Mishkin, B.C. Cooper, T.B. Lawton, N.K. Lay, and S.P. Katzmann, A Vision System for a Mars Rover, *Proc. SPIE Mobile Robots II*, vol. 852, 1987.

IMPACT OF END EFFECTOR TECHNOLOGY ON TELEMANIPULATION PERFORMANCE

A. K. Bejczy, Z. Szakaly and T. Ohm
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109

ABSTRACT

This paper is subdivided into three parts. In the first part generic requirements for end effector design are briefly summarized as derived from generic functional and operational requirements. Included in this part of the paper is a brief summary of terms and definitions related to end effector technology. The second part of the paper contains a brief overview of end effector technology work as JPL during the past ten years, with emphasis on the evolution of new mechanical, sensing and control capabilities of end effectors. The third and major part of the paper is devoted to the description of current end effector technology work at JPL. The ongoing work addresses mechanical, sensing and control details with emphasis on mechanical ruggedness, increased resolution in sensing, and close electronic and control integration with overall telemanipulator control system.

INTRODUCTION

Space operations planned in the next decade include assembly, servicing and repair of space systems. Some of these operations are expected to be performed by the use of teleoperators or telerobots. The difference between teleoperator and telerobot is the mode of control. A teleoperator is continuously controlled by a human operator in all activities. In contrast, a robotic system operates in automatic mode of control. A telerobot combines control elements of teleoperators and robots. A telerobot system permits both direct operator control and automatic control supervised by the operator.

The term "telemanipulation" used in the title of this paper signifies a remote manipulator system and its operation in both teleoperator and telerobotic modes of control, including all elements needed for the remote operation: the arm, hand, sensors, electronics, microprocessor, interfaces, base support, communication links, the control station with displays and with manual and computer control input devices.

Robot hands, or end effectors, are essential elements of telerobot systems to be employed in space since, in the proper since of the word, "manipulation" is the function of the hand. Using the analogy of the human arm-hand system, the arm is a positioning, orienting, power and information transmission device, while the hand is a powerful tool and delicate sensory organ. Dexterity and smartness in telemanipulation to a large extent resides in the capabilities of robot hands. End effector technology has a major impact on task performance in telemanipulation.

In the first part of the paper generic end effector functional requirements are outlined including terms and definitions. End effector technology work at JPL during the past ten years is briefly reviewed in the second part of the paper. In the third and major part of the paper ongoing end effector technology work at JPL is described.

REQUIREMENTS, TERMS AND DEFINITIONS

The general hand design requirements can be subdivided into four principal areas: (i) mechanism, (ii) sensing, data acquisition and transmission, (iii) control, and (iv) man-machine interface for decision and control.

Mechanism

The most important function a hand has to perform is to grasp and to hold objects. Even though this seems to be rather simple, one has to keep in mind that objects come in different sizes, weights and shapes and with many more characteristics to be considered such as fragility, ob-

ject presentation, space restrictions, accuracy, etc.

It is obvious that no single hand design can accommodate all requirements to successfully handle all objects. Even the most sophisticated end effector, the human hand, uses a variety of manual and power tools and still needs other aiding devices for even quite common tasks.

In searching for an answer of what comprises a useful robot hand from a mechanical point of view, the word "versatility" comes to ones mind: if the end effector can handle a large variety of different objects, it can be considered versatile. More sophistication will be gained if the hand is able to manipulate objects (i.e., to turn an object within the hand or pull the trigger of a hand-held drill press while holding it).

Employing tools was the turning point that changed early man's life. It will have the same effect on robot hands where the usage of tools will enhance the robotic capabilities and application ranges. But one hand alone cannot accomplish much by itself. Therefore, the final configuration of a robot hand system will be a multi-handed configuration permitting exchange of end effectors on a given arm.

Two types of hands need to be considered. The first is the one degree of freedom hand which can be made smart through incorporating a variety of different sensing capabilities built into the hand. Its mechanical design is relatively simple, thus reliable. But it is limited to grasping objects without manipulating them. The other type of smart hand is the dexterous hand with several fingers and finger joints. In this paper we only consider one-dimensional robot hands.

One-dimensional hands have to opposing fingers of some geometrical form that can clamp the workpiece in-between. Hand performance requirements for these hands can be established according to the required tasks to be accomplished. The capability to execute as many tasks as possible with one hand design will determine the hand's kinematic motion and shape. Should one hand not cover all intended applications, exchangeable plug-in end effectors might be considered, especially if the objects and loads vary greatly in size and shape.

It is usually desirable to have concave sections in the clamping surfaces to lock-in the object rather than relying on frictional forces alone. Hugging an object allows reduction in clamping force which might result in structural size reduction. Concave surfaces and other geometrical shapes also assist in grasping and centering of workpiece in the gripping area. It will aid the controller in recognizing if the object is properly grasped. The clamping force should be adjustable.

A linear closing motion is best suited for control purposes. Independent activation of each finger can aid in aligning the hand for grasping. Even better is a coasting capability while grappling so that the fingers can align at the object. Elastic elements or a spring system can be incorporated in the finger, assuring better clamping characteristics with a more gradual force application and reduced slippage between hand and workpiece.

Sensors

Intelligent operations require a great amount of sensory information which includes force, moment, position, tactile, temperature and proximity sensing, object recognition, global and local vision and many more. Space permitting, any number of sensors can be built into the hand. Much work is needed to downscale the sizes of sensors, for most of them are far too bulky for practical applications within or at the hand.

If possible, sensors and feedback routing should be placed entirely within the physical confinements of the hand for protection. Otherwise, contaminants and moisture inflow might hamper their operations or material handling may crush them if located in exposed positions. Tactile and any other sensors which are located on the surface need to be sealed and extremely rugged. The amount of sensory feedback will determine if local preprocessor are needed. Multiplexing will always be necessary with smart hands.

Control

Robots do not yet have the capability to adjust to major changing situations. A human operator is therefore required in the control loop to make all major control decisions. Artificial intelligence will eventually help but is still years away in its development. With human operators controlling the teleoperation system, the controller must present the pre-evaluated feedback to the operator in easy-to-understand form for quick recognition, comprehension and decision-making by the operator.

Man-Machine Interface

The information flow between the operator and the teleoperator system is a presen-

tation of sensed information to the operator and the operator's control decisions back to the controller.

With vision being the most important sense, a visual signal in the form of a mono or stereo TV picture will have to be transmitted to the operator from the mechanical hand. It will provide the operator with a sense for where the hand is reaching. Additional cameras might be mounted at the arms of the robot to aid in grasping. Other sensory information can be presented in graphic, acoustic or some other form that provides convenient state evaluation possibilities for the operator.

Mechanical, electromechanical and electrical interfaces are common in master-slave arrangements. Positional control will be simplified if the operator manually performs the motion which the end effector will repeat. This positional control can be done in a master-slave control arrangement. The master-slave arrangements incorporate backdriving (or force-reflecting) capability. This capability greatly enhances the operator's perception for control decisions.

General and specific end effector technology requirements are treated in more details in References 1 and 2.

PAST END EFFECTOR DEVELOPMENT AT JPL

The JPL end effector development adopted an evolutionary approach to generate important and needed capability increases stepwise. The basic idea was that the first generation smart hand models should be one degree-of-freedom (d.o.f.) parallel-claw end effectors equipped with proximity, tactile, six d.o.f. force-torque and one d.o.f. grasp force sensors. Several smart hands of this category have been developed during the past ten years. These prototype models differ in their end effector size, drive mechanism, claw shapes, load handling capacity, local electronics and control design, and subsystem interface instrumentation.

An early smart hand prototype is shown in Figure 1. Indicated on the figure are three sensors: a six d.o.f. wrist force-torque balance sensor, two proximity sensors in each claw, one pointing forward and one pointing downward, and a thirty-two-point touch sensor on each claw. Each touch-sensing spot in the gripping area is actually a copper pin. The contact pressure on the pin will cause the circuit underneath to close, generating a simple "on" signal. The center-to-center distance between the contact pins determines the touch sensing area resolution. The proximity sensors in Figure 1 have a distance sensing range of 4 inches with a resolution of 0.05 inches. More on this smart hand can be found in References 3 and 4.

Two smart hands are shown in Figures 2 and 3 developed for control performance evaluation using the simulated Space Shuttle Remote Manipulator System (RMS) at the Johnson Space Center (JSC). The four-claw end effector shown in Figure 2 is equipped with four proximity sensors with a distance sensing range of six inches. These sensors can measure range and pitch and yaw alignment errors. More on this experimental sensor system and on the performance results can be found in Reference 5. The end effector, which can have a four-claw or three-claw configuration and shown in Figure 3 is equipped with a force-torque sensor with a dynamic sensing range of 200 lbs, with 0.2 lbs resolution. The end effector assembly schematic clearly shows that the force-torque sensor frame is an integral part of the end effector mechanism. In fact, the whole mechanism is designed around the sensor frame. Note also in Figure 3 the local electronic instrumentation required to operate this system. More on this smart hand and on the experimental results can be found in References 6 and 7.

Figure 4 shows a smart hand designed for tests on an Orbiting Maneuvering Vehicle (OMV) Protoflight Manipulator Arm (PFMA) at the Marshall Space Flight Center (MSFC). The JPL-OMV smart hand is a one-d.o.f. gripper with intermeshing jaws consisting of parallel plates with a V groove center section. Thus, the claws can mechanically lock on square or cylindrical objects in two-d.o.f. The jaws can travel on a linear path while gripping, and their maximum opening at the tip is 6.5 cm. Each jaw has a built-in load cell to measure gripping force in the range of one to 600 Newtons. The jaws are driven by a DC motor via opposing lead screws. Double slides, supported at both ends for compactness and stiffness, guide the jaws' motion. Each slide is on a separate hardened and ground steel rod. A channel built into the drive system's frame gives additional guidance. The entire smart hand mechanism mounts to the robot arm wrist through a six-d.o.f. strain gauge load cell system by which the three interaction forces (Fx, Fy, Fz) and moments (Mx, My, Mz) with the environment are measured in the range of 120 Newtons and 70 Newton-Meters.

Self contained in sensor data acquisition, data processing and motor control, the JPL-OMV smart hand has three built-in

microprocessors (Motorola MC68701 and MC68705 units), as shown in Figure 4. Thus, the command interface, force-moment and position feedback to the remote support equipment require only a single full duplex RS-232 link. The distributed microprocessors' architecture in the hand uses advanced integrated circuits, including hybrid and high level multifunctional packages, thereby minimizing the chip counts. Custom design circular and annular printed circuit cards support the hand's controller ICs. Seven slip rings interface the local electronic circuits with the central electronics. Four are used for power transmission, two for bidirectional data communication, and the seventh for system ground.

Power for the motor and electronics comes from a support chassis that also houses a National Semiconductor 32016 microprocessor and a Parallax graphics processor for high level control and real-time force-moment graphics display. A control box is used to operate the hand, setting the gripper control mode, changing the give force, rate and position, and adjusting operating parameters such as force and rate limits. This gripper can handle fragile objects with a gentle grasp force of from one to five Newtons, or hold a tool with a firm grip of up to 600 Newtons.

Force and torque gripper control takes place in the hand itself, using a microprocessor for motor control. Commands from the control box are sent to the motor controller via a serial link and the communication processor. On this same route, force, moment and position information is continuously sent to the support chassis for graphic display on a TV monitor. The forces and moments measured by the six-d.o.f. strain gauge force-moment sensor assembly are represented as bar graphs in a star configuration which suggests a perspective view of the Cartesian reference frame of the gripper. Jaw opening and clamping force are represented by vertical bars on the left side of the graphics display. Software provides for two display adjustments; taking away unwanted load bias (like gravity) and scaling the display bars by specifying the force and moment level corresponding to a full bar-graph display. For performance evaluation of this JPL-OMV Smart Hand see Reference 8.

Figure 5 shows a smart hand developed for the Goddard Space Flight Center (GSFC) Flight Telerobotic Servicer (FTS) ground test facility. This hand is also equipped with six-d.o.f. force-torque and one-d.o.f. grasp force sensors. The operation of this smart hand is very similar to the operation of the JPL-OMV hand described above. Note, however, the reduced volume of electronics of this smart hand and the V-shaped grooves that contour the inner surface of the jaws in two perpendicular directions, as compared to the electronics and to the claw configuration of the JPL-OMV smart hand.

The smart hand shown in Figure 6 was designed at JPL to fit a medium size industrial robot arm such as the PUMA 560. It is used at JPL for research in hybrid motion and force modes of control. The hand has three parts: a jaw mechanism, sensors and local electronics. Powered by a DC torque motor through gears and recirculating ball spindles, the parallel jaw gripper mechanism moves on rails and is supported by linear bearings to minimize friction. Each jaw subassembly consists of three parts: a moving support, a grasp force sensor operating in the range of one to 150 Newtons, and an interchangeable jaw tip. As seen in the photo, V-shaped grooves contour the inner surface of the jaws in two perpendicular directions, assuring a friction-independent, mechanically firm grasp. This permits the gripper to mechanically lock on rectangular or cylindrical objects in two directions with two-d.o.f. constraints or to connect to a T-shaped tool head with three-d.o.f. constraints.

Behind the base of the jaws is a six-axis force-moment sensor with a dynamic range of 75 Newtons and 20 Newton-meters for reading the three orthogonal forces and moments induced by the robot hand's interaction with the environment. This sensor consists of a Maltese cross-like structure instrumented with strain gauges. Strain gauge readings from this sensor are acquired by the local microprocessor, formatted, and transmitted to the central control computer. There, control programs are executed and sensor data are sent to a remote control station.

Local electronics for this smart hand are housed in a shell attached to the force-moment sensor and connected to the robot wrist. In it are two custom printed circuit boards, one for the digital and one for the analog input/output electronics. The digital electronics are based on an Intel 8097 microprocessor with a high number of built-in functions that permit effective management of the real-time multi-tasking environment. The local software system consists of a background process for message analysis and message generation, and an interrupt driven routine for the real-time functions of the controller. The microprocessor clock generates an interrupt every two milliseconds. Presently, three separate grasp control loops are implemented; position,

rate and force controls. When in force control mode, the controller maintains a preset grasp force until the central control computer issues a different command.

More on this smart hand can be found in Reference 9. This hand, called Model A PUMA Smart Hand, is presently being used on one of the PUMA 560 robot arms in the JPL laboratory breadboard system for dual-arm teleoperation described elsewhere in this proceedings. (See Reference 10.)

CURRENT SMART HAND DEVELOPMENT AT JPL

The ongoing end effector technology work at JPL is concentrated on the redesign of Model A PUMA Smart Hand to obtain wider dynamic range in task performance both mechanically and electronically. This new design, called Model B and Model C PUMA Smart Hand, contains numerous novelties which fall into two categories: electronic novelties and mechanical novelties.

Electronic Novelties

Figure 7 shows Model A and Model B PUMA smart hands side by side. The electronic novelties included in Model B smart hand (and also in Model C smart hand the mechanism of which is described later in this paper) are the following:

- Instead of a conventicnal design where a microprocessor performs the data collection and communication functions, this new electronics employs a high speed custom designed state machine. This state machine interfaces to a bidirectional fiber optic link for high speed data communication. This circuit achieves a factor of 100 improvement in data collection speed and servo rates up to 10 kHz for a 16 input system. The high servo rate makes it possible to perform advanced signal processing on the force data.
- Due to the high bandwidth of the optical communication link it is not necessary to process the data locally in the hand. All data processing functions are performed at the host processor so all of the software can be written in a familiar and convenient development environment. The software can be changed much more easily.
- In a conventional system the strain gauges are excited by a DC voltage around 5 to 10 volts. The higher this voltage the more signal we get out of the strain gauges with a constant noise level. The voltage is limited by the heat produced in the strain gauges. This voltage is typically not more than 12 volts. Our new electronics uses a narrow pulse (5 μsec wide) to excite the gauges and a very high voltage (100 V). This results a factor of 10 improvement in the signal to noise ratio of the force measurements.
- The above mentioned high strain gauge voltage is variable by software controlling the full scale force range. This method keeps the 12-bit accuracy no matter what force range is used. The control range is a factor of ten, resulting in a virtual floating point force measurement system. The outcome is equivalent to having a 15-bit force reading at 5 kHz rate which can be processed to get a 17-bit value at the system servo rate of 1000 Hz.
- When converting the 8 raw force readings to the 6 Cartesian forces and torques, four of the output numbers are computed as differences of two of the raw readings. If one of the two numbers subtracted reach saturation due to a large force on some other axis, the difference will be inaccurate. To avoid this situation the new circuit subtracts these two numbers in hardware. The result is an accurate Cartesian reading of a small force/torque even if there are large forces acting on other axis. In this arrangement we have 12 raw readings that are converted to 6 Cartesian forces and torques.

According to the block diagram shown in Figure 8, the end effector electronics consists of the following major subsystems:

- PLL clock and state machine
- Power supply
- Motor drive
- A/D converter and input multiplexer
- Sample and hold with preamplifier circuits

The functions of these are as follows:

- The PLL clock and state machine block converts the serial data coming in from the host processor into parallel data bytes and words written to the internal data bus. When all expected bytes have come in, the state machine switches to transmit mode and converts the parallel data coming from the internal bus to a serial bit stream that is transmitted on the output optical fiber to the host processor. The output data also includes the entire received input data as an echo for debugging purposes. This block consists of the following pieces:

- Edge detector
- Packet detector
- Fast clock
- PLL state machine
- Extra bit removal logic
- Serial to parallel conversion logic
- Read/write pulse generation logic
- State change logic

- The function of the edge detector logic is to generate a 10 nanosecond wide pulse every time a positive edge is detected in the data. The packet detector generates a reset signal if there is no input data coming in, removes the reset signal as soon as data begins to come in. The fast clock is an accurate time base for the entire system. This clock runs at 24 MHz, 8 times the bit rate.

- The PLL state machine generates a 3 MHz two-phase clock for data decoding purposes. This clock is phase locked to the pulses coming from the edge detector.

- The extra bit removal logic removes every fifth bit from the incoming data stream. Upon transmission every four bits of data is followed by an extra bit which is added in order to guarantee a level transition that keeps the PLL clock synchronized. These extra bits have to be removed from the receiver data.

- The serial to parallel conversion logic generates the data bus signals from the incoming serial data and it generates the serial outgoing data from the parallel bus signals.

- The read-write pulse generation logic generates a write pulse every time a full byte or word appears on the data bus. This logic supports up to 256 devices on the data bus. When transmission is performed, this logic generates a read pulse for every byte or word to be read into the parallel to serial conversion logic.

- The state change logic counts the incoming data bytes and after a preset number of bytes have come in, it switches to transmit mode. Later when the preset number of bytes have been transmitted, it switches off all circuits and returns to idle mode.

The function of the power supply is to generate the following supply voltages:

± 5V
± 15V
± 50V

from a single 30V supply coming into the hand. The ±50V supply is not DC but it is a pulse instead. This pulse is emitted every time the hand goes from idle to receive mode and is used to excite the strain gauges. The size of this pulse is under software control. It can be varied from 5 to 50 volts.

The motor drive consists of two identical output circuits. Each one can be software controlled to produce a voltage from -15 to +15 volts. The motor is connected between these two outputs as a bridge. Thus, the motor voltage can vary from 0 to + or - 30 volts.

The A/D converter is a successive approximation 12 bit unit. It performs one conversion in 3 microseconds. The input to this A/D converter is unusual in the sense that the sample and hold circuits are in front of the input multiplexer and so one for each input is needed. This arrangement makes it possible to sample all of the inputs simultaneously, improving the signal quality. This arrangement is also needed because all of the strain gauges are excited with the same pulse. The A/D converter section also includes a standard voltage reference.

Sample and hold with preamplifier circuits. There are 16 input circuits of which 12 are equipped with local D/A converters. These D/A converters are under software control, they are used to remove any offset from the data. Such an offset varies depending on hand orientation, the object grasped and the distance of the grasping from the FT sensor center point. By locally removing these potentially large offsets, the sensitivity of the hand is substantially enhanced. The remaining 4 inputs that do not have D/A converters are used for finger position, motor current and supply voltage sensing.

This hand cannot exist on its own, it always has to be examined in relation to the control processor that it is connected to. The control processor has to be equipped with a matching fiber optic link. This link has been developed to allow our processors to communicate to each other. Currently this link only exists for the Intel iSBX bus of the 32016 within the MULTIBUS environment, but within a few months we are going to make a version for the 68020 in the VME bus environment. The controlling processor outputs a packet to the hand that contains the 16 output commands. Two of these define the motor drive current, the rest specify the bias values for 12 of

the A/D inputs. The hand responds with an echo of these same values followed by the 16 A/D readings, two bytes each. It is up to the control processor to perform the following functions:

- Finger motion control, such as force servicing.
- Signal processing of the input data, such as noise elimination.
- Coordinate transformation of Cartesian forces to task frame.

Although the top speed of the hand is around 10,000 Hz, currently we perform the above functions at a 5000/1000 Hz servo rate. The force readings are taken at 5000 Hz, and the output filtered force data is computed at a 1000 Hz rate.

Mechanical Novelties

The latest smart hand mechanical design at JPL is known as the Model C PUMA Smart Hand. This model stands 8.5 inches from its mounting plate to its fingertips and spans just over 7 inches along its widest point (see Figure 9). Physically, the hand attaches to the manipulator arm at the base of a cylindrical bell which houses the four electronic boards described above under Electronic Novelties. Mounted to the upper end of the bell is a force/torque sensor which, in turn, attaches to the mechanism structure. This component houses the motor and drive system to activate the fingers. The two fingers protrude from this structure, being attached via grip force sensors. These fingers, designed to grasp both flat and round objects, are capable of handling up to 3.6 inch objects with a maximum grip force of 60 lbs.

Mechanically, there are three areas that were designed based on the criteria submitted. These are: 1) the overall structural design, 2) the drive mechanism, and 3) the sensing elements. The structural design involved creating a lightweight yet rugged instrument that would take the abuse submitted in a laboratory environment. Designing the drive mechanism consisted of developing a durable, dependable transmission system to actuate the fingers. Sensing design encompasses the detection of loads at the worksite. Each of these will be discussed.

The entire structure of the hand is of anodized aluminum alloy. To create a rugged yet lightweight construction, aluminum 7075-T6 is used since its high strength allows thinner cross-sections. Structurally, the hand is designed to handle a 50 lb external force and 50 ft-lbs of external torque, while only weighing between 4.5 and 5 lbs (total predicted weight with electronics). Considerable care was taken to shield various delicate components (such as the electronics) from being damaged, but still provide easy access for servicing.

The fingers are actuated by a brushed, direct current, frameless motor (manufactured by Magnetic Technology, capable of 110 oz-in of torque) which directly drives a leadscrew on which the fingers follow. The fingers are supported on Schneeberger crossed-roller linear bearings. To create the opposing motions of the two fingers, the motor was mounted at the center of the leadscrew with right- hand threads extending from one side while left-hand threads are on the other. The result is that the fingers will move in opposite directions for a given motor rotation. This, coupled with the leadscrew's high mechanical advantage, is a very simple and reliable transmission system which provides a compact conversion of the motor's angular motion into the finger's required high-force, linear motion.

Using this type of drive system basically resulted in deciding what type of nut and leadscrew assembly to use, since the losses and mechanical advantage of this assembly dictate the motor size. All of JPL's previous designs which incorporated a leadscrew drive used a ball nut assembly. A ball nut is basically a ball bearing whose races are the screw threads. Such assemblies exhibit very low friction and therefore are highly efficient (greater than 90% as opposed to about 15 to 25% for a bronze nut on a steel leadscrew). Since the efficiency is over 50%, these assemblies are also backdriveable. Three major problems are that ball nut assemblies are susceptible to dirt and debris, require precise alignments, and are very expensive since it would have to be custom-made for this hand (costing about $5000 per assembly). This led to researching alternative leadscrew designs.

Analyzing the mechanics of leadscrews resulted in four basic design conclusions to improve performance: 1) the screw diameter should be as small as possible, 2) the coefficient of friction between the nut and screw should be below 0.1, 3) the lead angle should be as high as possible (up to 45 degrees), and 4) a square thread should be used as opposed to a acme or "V" thread. Using a square thread increases efficiency and reduces problems caused by dirt because it tends to clean the thread during operation. The first and third conclusions are dependent upon the loading criteria and physical limitations of the hand. The second, though, is primarily a function

of the materials chosen for the nut and leadscrew. Further research found that the coefficient of friction value was the single most contributing factor to having a high efficiency system, as opposed to altering the screw design to reduce frictional effects.

Concentrating the research on various types of low-friction materials, several possible solutions were found for the nut material to operate on a stainless steel leadscrew: 1) Teflon, 2) a solid-film lubricant over a base metal, and 3) Teflon filled Delrin. Teflon is the obvious choice because it has the lowest coefficient of friction of any known solid (about 0.04), but this polymer may have a problem of "creeping" when under a sustained load (while gripping). Literature regarding solid-film lubricants indicated that they may have friction levels comparable to Teflon, but test samples showed this not to be the case (friction about 0.15 to 0.20 at the operating load). DuPont manufactures a Teflon filled Delrin (acetal resin) which would not have the creep problems of the Teflon, but the friction would be twice as high (about 0.08). Overall, Teflon would be the best solution if the creep is acceptable. In the case that Teflon should prove to be unacceptable, a Teflon filled Delrin will be used.

With this drive transmission system, the mechanism would operate at about 55% efficiency with Teflon nuts as opposed to about 40% with the Delrin nuts. Neither system will be backdriveable, even though the Teflon system operates over 50%. This is because the motor's cogging torque is high enough to prevent backdriving at the rated load (60 lbs). This results in a possibly desirable feature. This hand will be capable of gripping an object and maintaining the grip force without continuously supplying power to the motor, yet only incorporate a minimal amount of friction if servoing is required. The accuracy of grip force magnitude after the motor power is discontinued will be a function of the system's and the gripped object's compliance since there will be a slight mechanical relaxation. Although any external loads which are applied may result in undesirable finger forces, this mode could be useful for moving or holding an object in free space with no power dissipation.

Loading and position sensors are essential for completing tasks efficiently. The Model "C" is equipped with three such sensors: 1) finger position sensor, 2) grip force sensor (GFS), and 3) force/torque sensor (FTS). For this hand, it was decided that the finger position did not need to be known very precisely. Therefore a linear potentiometer is used for this purpose.

To measure the forces applied to the finger, a GFS is used. This sensor is part of the hand's structure which connects the fingers to the mechanism. Any finger force which is applied must be transmitted through this structure, thus causing it to deflect. Semiconductor strain gages are used to detect this deflection and thereby measuring the grip force.

The GFS's shape is that of a rectangular, tubular box (see Figure 10). This design has two key features. Firstly, when under load, the sensor deforms similar to a four-bar linkage, keeping top parallel to the bottom. This results in the faces of the two fingers remaining parallel within a designed tolerance. Secondly, by placing the strain gages in specific locations, the effects of applied moment can be cancelled, thereby measuring only the shear force. The shear force is equal to the grip force whereas the moment is due to where the force is applied on the finger (how far from the sensor). Through a detailed theoretical analysis which was verified by a series of tests, the test location determined for the gages was with all four (full bridge) on the same outer face (see Figure 10). This configuration resulted in the most accurate readings and also provided for the easiest installation.

Between the mechanism and the electronics bell is where the external forces and torques are detected with the FTS. The FTS also uses strain gages, but the structure is much more complex than the GFS. It basically consists of two rigid bodies connected with four beams (see Figure 11). One body attaches to the mechanism while the other to the bell. Therefore any external forces or torques must pass through the beams, resulting in deflections which are detected by 32 strain gages (8 full bridges, 8 gages on each beam). Through the proper decoding scheme, the forces and torque about three orthogonal axes can be determined.

Overall, the Model "C" design has very desirable features for a hand of this nature. It is very rugged in its lightweight, high-strength construction. The actuation system is very simple, incorporating a minimal number of moving parts. Furthermore, the design is very reliable mechanically, which is implied by the simplicity of the transmission system. The Model C smart hand will also be used at the Intelligent Systems Research Lab (ISRL) of Langley Research Center (LaRC).

CONCLUSION

JPL's smart hands represent only the beginning of the evolutionary trail. Future plans include the addition of new electro-optical proximity and tactile sensing capabilities. Used in close-up work, optical sensors beam infrared light at the object of interest. Reflections from the object's surface will be triangulated to provide depth information. Tactile sensitivity will give robot hands abilities similar to those of human skin, with its sensitivity to touch.

The trend to develop smarter robot hands challenges mechanical design and sensor and microelectronics technology. Hands such as those at JPL were inconceivable just a few years ago, due to the bulk of the local controlling electronics. As circuit size continues to shrink, smart hands will get brighter, bringing increased benefits both in space and on earth.

ACKNOWLEDGMENTS

The Model A, B and C mechanical design is by Z. Vigh and analysis by T. Ohm. The Model B and C electronics design is by Z. Szakaly. The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

REFERENCES

1. Bejczy, A., Jau, B., "Smart Mechanical Hands for Teleoperation in Earth Orbit", Proc. of the 19th Aerospace Mechanisms Symposium, NASA Ames Research Center, May 1-3, 1985.

2. Mishkin, A.H., Jau, B.M., "Space-Based Multifunctional End Effector Systems," JPL Publication 88-16, April 15, 1988.

3. Bejczy, A., "Smart hand - Manipulator Control Through Sensory Feedback", JPL D-107, January 15, 1983.

4. Bejczy, A.K., and Vuskovic, M., "An Interactive Manipulator Control System," Proc. of Second International Symp. on Mini- and Microcomputers in Control, Fort Lauderdale, Fl., Dec. 10-11, 1979.

5. Bejczy, A.K., Brown, J.W., and Lewis, J.L., "Evaluation of Smart Sensor Displays for Multidimensional Precision Control of Space Shuttle Remote Manipulator," Proc. of 16th Annual Conference on Manual Control, MIT, Cambridge, MA., May 5-7, 1980.

6. Bejczy, A.K., and Dotson, R.S., "A Force-Torque Sensing and Display System for Large Robot Arms," Proc. of IEEE Southeastern '82, Destin, Fl., April 4-7, 1982.

7. Bejczy, A.K., Dotson, R.S., Brown, J.W., and Lewis, J.L., "Manual Control of Manipulator Forces and Torques Using Graphic Display," Proc. of IEEE International Conference on Cybernetics and Society, Seattle, WA, Oct. 28-30, 1982.

8. Hannaford, B., "Task-level Testing for the JPL-OMV Smart End Effector", Proceedings of the Workshop on Space Telerobotics, Pasadena, CA, January 30, 1987, JPL Publication 87-13, Vol. 2, pp. 371-380.

9. Fiorini, P., "A Versatile Hand for Manipulators," IEEE Control Systems Magazine, Vol. 8, No. 5, Oct. 1988, pp. 20-24.

10. Bejczy, A.K., Szakaly, Z., and Kim, W.S., "A Laboratory Breadboard System for Dual-Arm Teleoperation," - see elsewhere in this Proceedings.

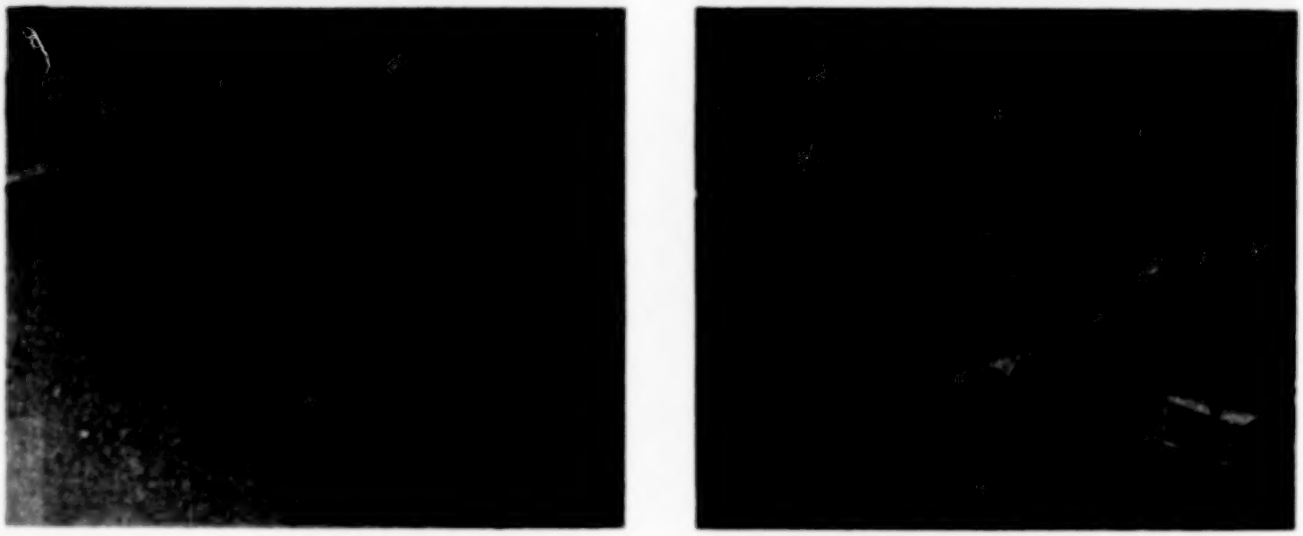


Figure 1. A Smart Hand Early Prototype at JPL (1976-80)

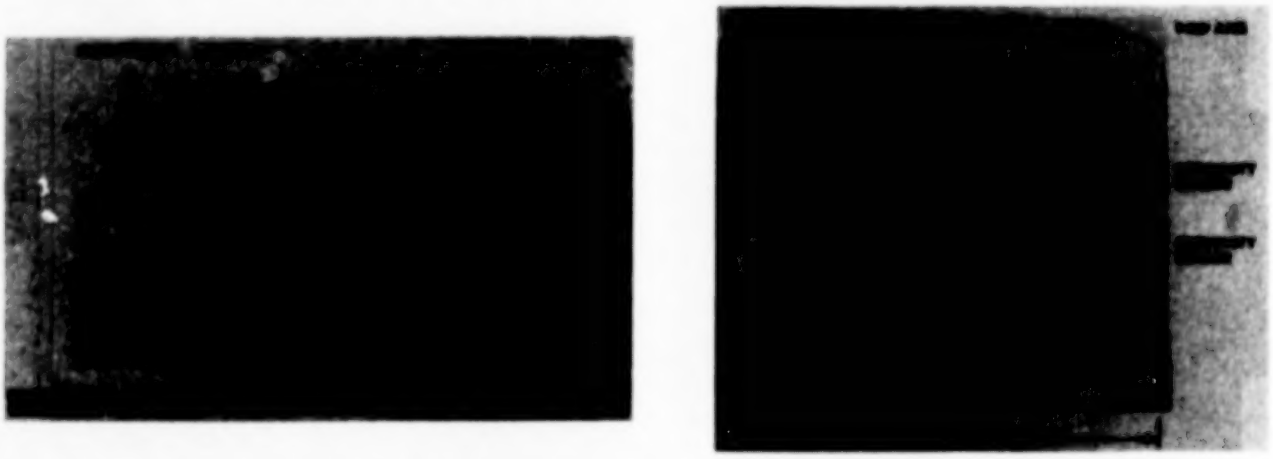


Figure 2. JPL-RMS Smart Hand with Proximity Sensors (1979-81)

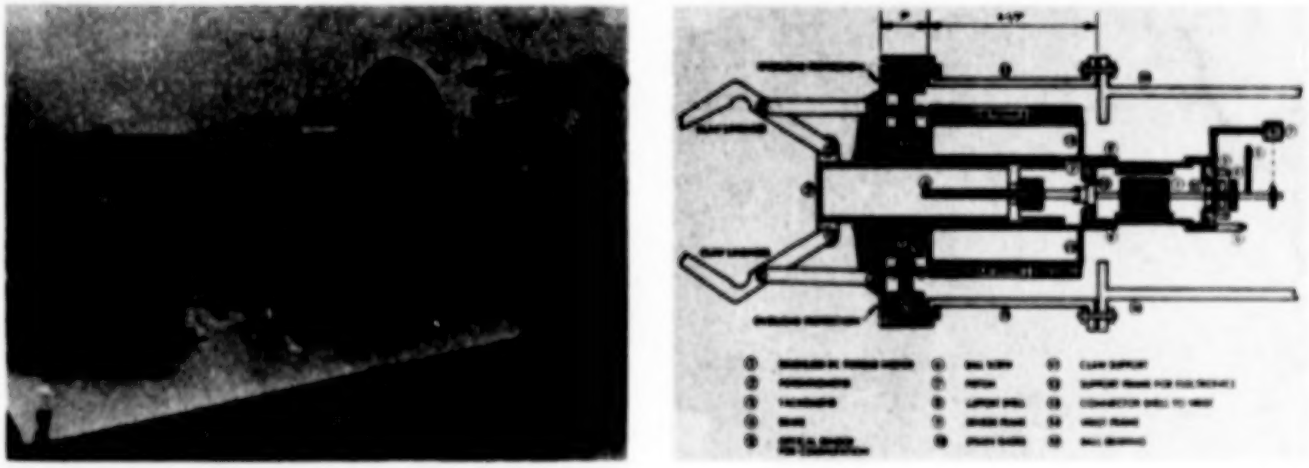


Figure 3. JPL-RMS Smart Hand with Force-Torque Sensor (1982-84)

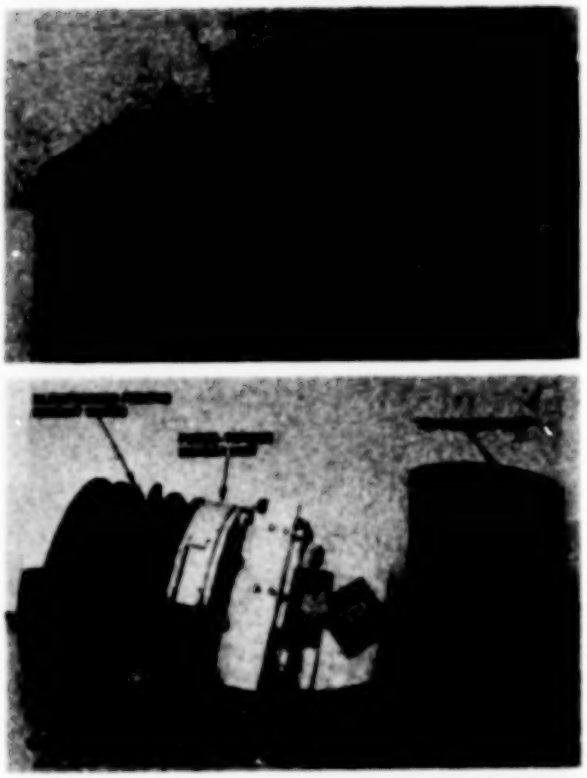
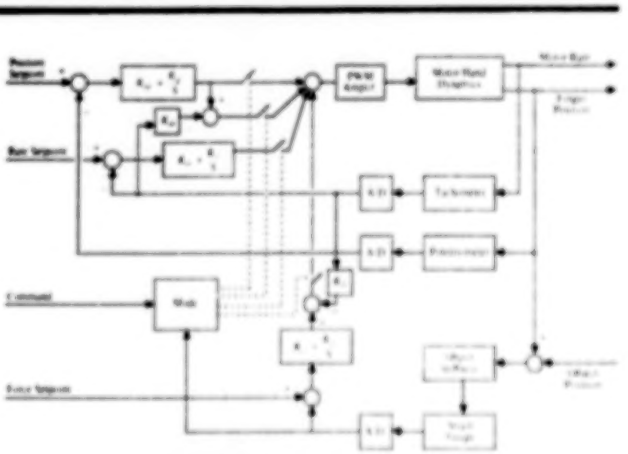
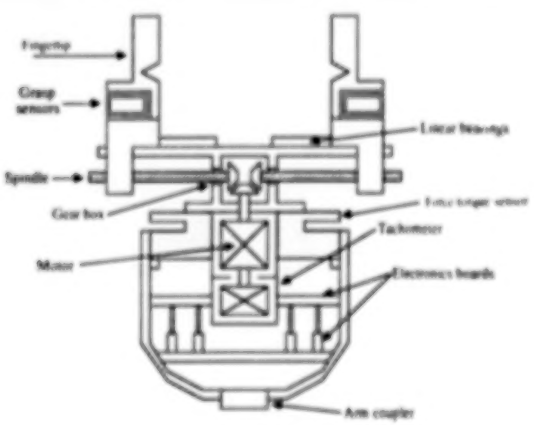


Figure 4. JPL-OMV Smart Hand (1984-86)



Figure 5. JPL-FTS Smart Hand (1987)





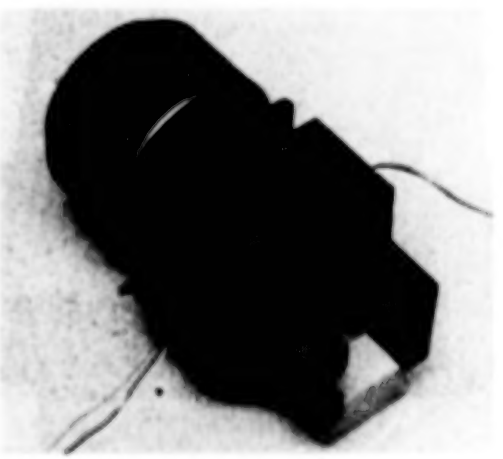


Figure 6. JPL-PUMA 560 Smart Hand and Control, Model A (1988)

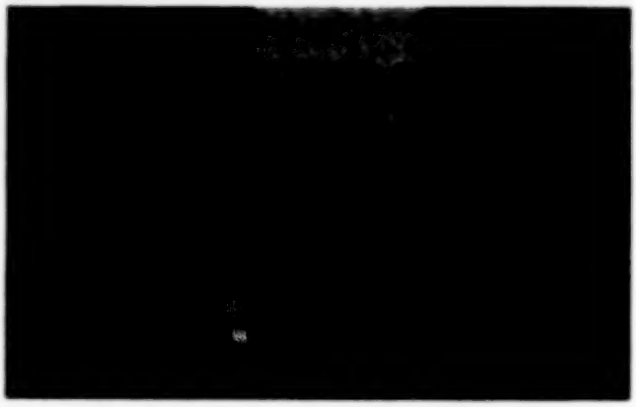


Figure 7. JPL-PUMA 560 Smart Hand Model B (1989)

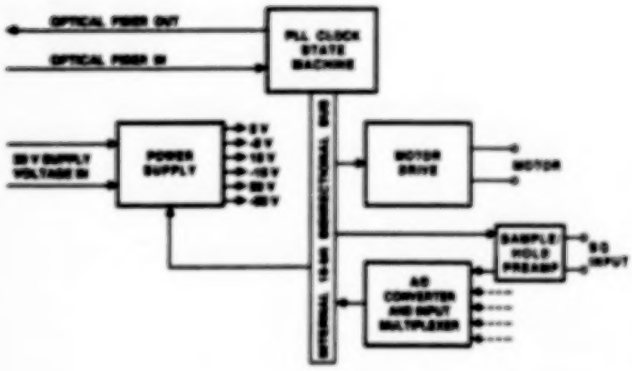


Figure 8. JPL-PUMA 560 Smart Hand Model B and C Electronics Block Diagram (1989)

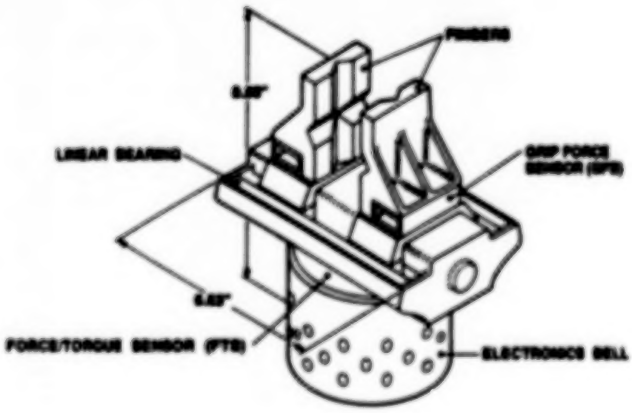


Figure 9. JPL-PUMA Smart Hand Model C Mechanism (1989)

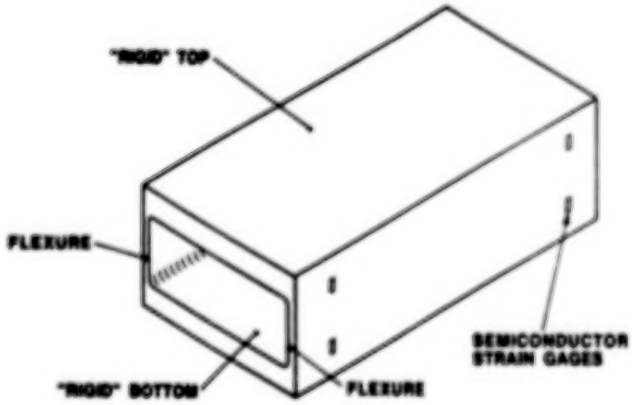


Figure 10. Grip Force Sensor Frame and Gages Schematics

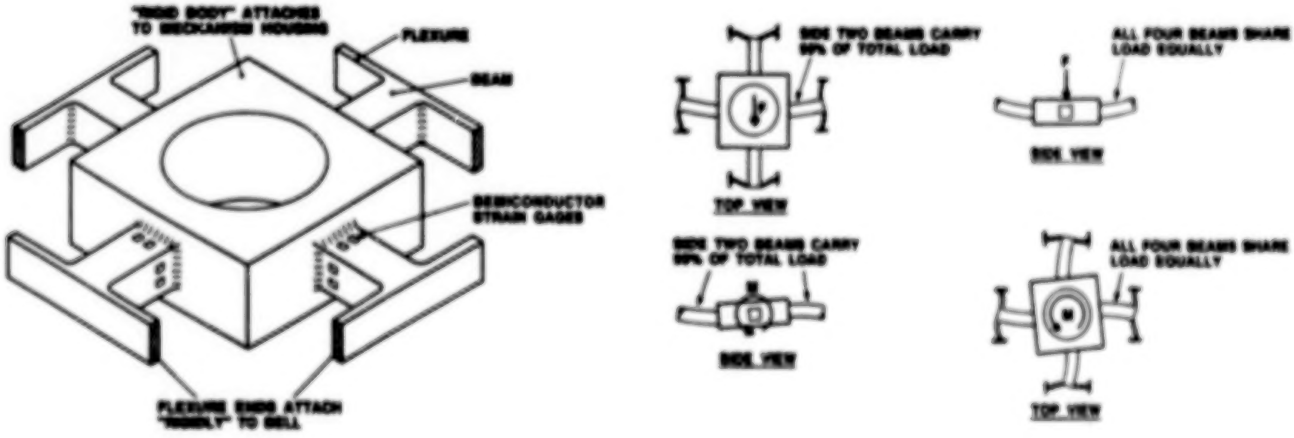


Figure 11. Force-Torque Sensor Frame, Gages and Loading Schematics

SMART HANDS FOR THE EVA RETRIEVER

Clifford W. Hess and Larry C. Li
NASA Lyndon B. Johnson Space Center
Houston, Texas 77058

ABSTRACT

Dexterous, robotic hands are required for the extravehicular actvity retriever (EVAR) system being developed by the NASA Johnson Space Center (JSC). These hands, as part of the EVAR system, must be able to grasp objects autonomously and securely which inadvertently separate from the Space Station. Development of the required hands was initiated in 1987. This paper outlines the hand development activities, including design considerations, progress to date, and future plans. Several types of dexterous hands that were evaluated, along with a proximity-sensing capability that was developed to initiate a reflexive, adaptive grasp, are described. The evaluations resulted in the design and fabrication of a 6-degree-of-freedom (DOF) hand that has two fingers and a thumb arranged in an anthropomorphic configuration. Finger joint force and position sensors are included in the design, as well as infrared proximity sensors which allow initiation of the grasp sequence when an object is detected within the grasp envelope. This hand design is being integrated with the JSC EVAR test bed and will be evaluated in the near future.

INTRODUCTION

Future space operations could benefit from a highly autonomous robot that is capable of assisting an extravehicular activity (EVA) crewmember. In addition to providing increased productivity for a given human crew complement, highly autonomous robots could also be substituted for the human crewmembers for certain hazardous EVA tasks. As a first step in the evolution of this type of robot, the Engineering Directorate at Johnson Space Center (JSC) has undertaken a project to develop the EVA retriever (EVAR), which is a highly autonomous, free-flying robot that is intended to retrieve items which inadvertently separate from the Space Station Freedom [1] [2] [3]. Figure 1 illustrates the retrieval activity. This project is currently in the ground demonstration stage, which consists of developing a test bed version of the EVAR system to investigate requirements and issues associated with building a flight-rated system that can be flown on the Space Shuttle as a flight experiment.

A key element of the EVAR system is the Smart Hand. A Smart Hand is a dexterous robotic hand that will allow autonomous grasping operations much like an EVA crewmember would use when retrieving an item. This type of hand will allow a dexterous grasping capability to be established and demonstrated for the EVAR that can then be evolved in the future to provide manipulative capability for follow-on generations of robots. This paper outlines the EVAR Smart Hand development, including objectives, approach, implementation, and future plans.

OBJECTIVES

The development effort is focused on two objectives: (1) to develop advanced technologies required to provide autonomous, adaptive grasping capability for the EVAR, and (2) to develop a ground demonstration of the Smart Hand concept and capabilities.

Based on operating conditions and interface requirements, the following six major performance goals are desired:

(1) Utilize ability to grasp objects without any prior knowledge of them - If the separated item is defaced or broken, a model-based grasp algorithm which requires prior knowledge of the item would most likely fail. Therefore, a non-model based approach is required.

(2) Utilize autonomous, adaptive grasping capability - The grasp action should be adaptive and autonomous, requiring only high-level commands to initiate the grasp action. This will free the central computer from having to perform low-level control tasks.

(3) Utilize same tool and equipment interfaces used by an EVA crewmember - By utilizing the same tool and equipment interfaces as used by an EVA crewmember, the need to develop special tool and equipment interfaces for the robot is minimized.

(4) Approximate EVA crewmember hand size and dexterity - By designing the robotic hand to have the same size and dexterity as that of an

EVA crewmember, the hand is compatible with the interface specified in (3).

(5) Maximize reliability - High reliability for the robotic hand contributes to a greater potential of mission success and also reduces the need for difficult, expensive EVA maintenance and repair.

(6) Minimize mass - Excessive mass in the robotic hand can create a large inertial load for the motors in the arms and hands, thereby reducing the dynamic response of the robot. Minimizing the mass will also reduce the propulsion fuel requirement. Since the EVAR is propelled by the manned maneuvering unit (MMU), minimizing the hand mass will also reduce the MMU propellant requirement.



Figure 1.- EVAR concept.

APPROACH

The approach taken is as follows:

- Procure and evaluate commercially available, state-of-the-art dexterous robotic hands technologies and develop in-house expertise. By understanding the features of existing hands, it was not necessary to re-invent the technologies already developed by other institutions.

- Develop and evaluate robotic hands in-house. In-house design concepts were fabricated and evaluated to provide alternate hand configurations that were compared with the commercially available robotic hands.

- Develop and evaluate sensors and control systems. Sensors and control systems are two important components in an autonomous robotic hand system. Advanced sensor system were also developed for different hands to provide position and force controls.

- Derive an optimized EVAR Smart Hand design based on evaluation results and performance goals. Each robotic hand was evaluated with the performance goals in mind. A design which came closest to meeting the performance goals was developed by incorporating features found in commercial and in-house developed robotic hands.

- Integrate prototype hands with the EVAR test bed system and demonstrate the EVAR Smart Hand capability to operate in a dynamic environment as part of an integrated system. When completed, this activity will provide an understanding of the multiple-body dynamics and hand-arm coordination associated with the EVAR scenario.

DISCUSSION

The following discussion describes the implementation of the approach outlined above.

A. Procure and evaluate commercially available robotic hands

Two commercially available dexterous robotic hands were procured for evaluation: the Utah/Massachusetts Institute of Technology (MIT) hand and the Stanford/Jet Propulsion Laboratory (JPL) hand. These two hands represent the state-of-the-art in dexterous hand capabilities. The capabilities of each hand are described below.

Utah/MIT Hand

The Utah/MIT hand is the most dexterous hand in the spectrum of hands available for our evaluation. The Utah/MIT hand system is shown in Figure 2. The hand has 16 degrees-of-freedom (DOF) arranged in an anthropomorphic configuration of three fingers and a thumb. The fingers and the thumb each have 4 DOF. Thirty-two pneumatic actuators operating at pressures up to 80 psi provide power to the hand. Tendons are used to transmit power from these pneumatic actuators to the joints through a system of pulleys and linkages called a "remotizer." Each joint is controlled by a pair of antagonistic tendons. Located inside each joint is a linear Hall effect sensor that measures the joint angles. Hall effect sensors are also located in the wrist to monitor the tendon tensions. A control box containing analog feedback control circuitry provides manual control of each joint with an interface for computer control that can be used in lieu of manual control [4].

Stanford/JPL Hand

The Stanford/JPL hand, designed by Dr. J. Kenneth Salisbury, is a 9-DOF hand with a non-anthropomorphic finger configuration and a large

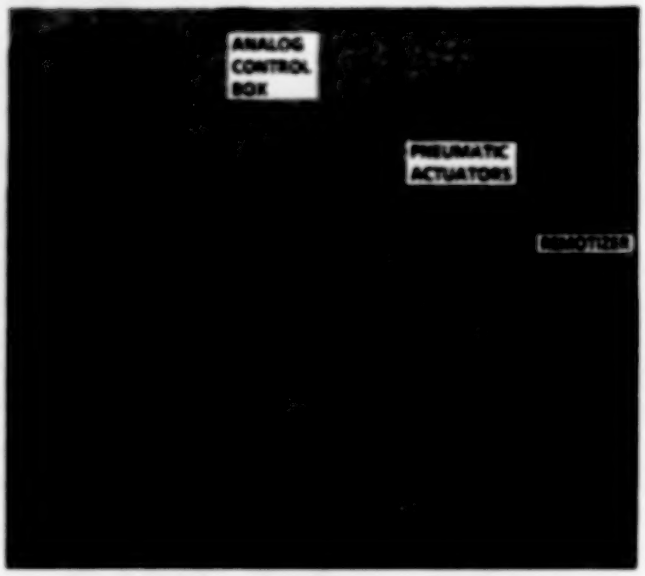


Figure 2.- Utah/MIT dexterous robotic hand system.

envelope of excursion. The hand has three fingers, each with three joints. The joints are driven by a set of metal cables that transmits mechanical power from 12 remotely located direct current (DC) motors equipped with position encoders. Located behind the proximal joint of each finger are four strain gauges that measure the cable tensions. The tension signal may be translated into a joint torque signal which is used in the servo control. The fingertips of the Stanford/JPL hand are made of a highly compliant material that provides the friction contact necessary for a secure grasp. Figure 3 shows the Stanford/JPL hand and its remote motor package.



Figure 3.- Stanford/JPL hand system.

B. Develop and evaluate robotic hands in-house

The NASA/JSC Crew and Thermal Systems Division (CTSD) I hand and the CTSD II hand were both developed by CTSD to evaluate in-house design concepts. The CTSD I hand was built in 1987 to support the Phase I EVAR ground demonstration system. The CTSD II hand will replace the CTSD I hand on the Phase II EVAR ground demonstration system. Also available for evaluation is the Direct Link Prehensor, a human-operated hand originally developed by NASA/Ames Research Center (ARC) and Stanford University for space suit applications. It has two fingers and a thumb in an anthropomorphic arrangement. A modified version of it was built at JSC. The important features of these hands are described below.

CTSD I Hand

The CTSD I hand has three fingers driven by a single DC motor. The three fingers are spaced 120 degrees apart, and they open and close simultaneously. Each finger contains three sections connected by joints. The sections are coupled by direct linkages; therefore, the push-pull motion created by the rod inside the proximal finger section will cause the other sections to move also. As the fingers begin to close, the distal finger section will bend around the object and trap the object within the grip of the hand for a secure grasp. The motions of the three fingers are also coupled by a cable-pulley system, so when any one finger is forced to stop, the other two will continue to close until all three fingers have stopped. These mechanisms are shown in Figure 4. Although this hand is a step beyond the simple parallel jaw gripper, it still has some drawbacks. The hand does not have enough independently controlled, articulated joints to allow alternate grasp arrangements, and it lacks sufficient sensory feedback to provide adequate information about the grasp quality.



Figure 4.- CTSD I hand.

CTSD II Hand

Like its predecessor, the CTSD II hand also has three fingers. However, there are several important differences. The fingers of the CTSD II hand are arranged in a two-opposing-one configuration to provide parallel grasping surfaces. This finger configuration is able to adapt to different shapes of objects better than the CTSD I hand configuration. The CTSD II hand is also designed with modular fingers. If additional fingers are required, they may be added without creating an impact on the overall design. Each finger is driven by one DC motor contained within the finger module. Tactile sensors and strain

gauges on each finger provide added sensory feedback [5] [6]. Silicon pads cover the tactile sensors for protection and provide a compliant, friction surface for a more secure grasp. The maximum amount of force each finger can exert is controlled by current-limiting circuitry in the control electronics.

A 2-DOF wrist has been designed to complement the CTSD II hand. With its two drive motors and differential gearing, the wrist is capable of simultaneous pitch and yaw. Encoders are mounted on the wrist motors for position feedback. Together with the CTSD II hand, the hand-wrist combination provides a total of 5 DOF and a fairly large working envelope. The hand and the wrist are currently being integrated with the Phase II EVAR ground demonstration unit for dynamic evaluation. Figure 5 shows the CTSD II hand-wrist-forearm assembly.



Figure 5.- CTSD II hand-wrist-forearm assembly.

Direct Link Prehensor

The Direct Link Prehensor, as shown in Figure 6, was originally developed by NASA/ARC and Stanford University to function as a space suit end effector that fits over the hand like a glove. The prehensor has a total of 6 DOF in an anthropomorphic configuration. It has two fingers and a thumb, with the thumb opposing the two fingers at an angle to provide grasping capability as well as some manipulation capability. The mechanical fingers are directly coupled to their human counterparts through a mechanical linkage system. The prehensor has been flown on the NASA KC-135 aircraft to evaluate grasping in a weightless environment using a mechanical hand [7]. The result of this study helped to focus the EVAR Smart Hand development by providing data on the speed and sensing requirements as well as assisting in the planning of grasp strategies. The prehensor has also been a valuable tool in tactile sensor development. Tactile sensors have been integrated with the prehensor to study sensor characteristics, sensor installation techniques, and selection of sensor sites.

C. Develop and evaluate sensor and control systems

Proximity Sensor

The concept of adaptive grasping has been demonstrated with the Utah/MIT hand using position, force, and proximity sensors. Two layers of proximity sensors have been installed on the Utah/MIT hand. The first layer consists of one reflective infrared (IR) sensor mounted in each fingertip. Sensors in this layer provide approach



Figure 6.- Direct-Link prehensor.

axis alignment as the robotic hand approaches an object during grasping. The second layer consists of the same type of sensors mounted in the middle segment of each finger, and they are used to trigger the closing of hand. Figure 7 illustrates the adaptive grasping concept. The proximity sensors currently being used are miniature reflective IR sensors made by TRW. Each sensor, as shown in Figure 8, has a transmitter and a receiver colocated on a chip with an area of 0.75 cm × 0.75 cm. The small size allows them to be installed on most robotic hands. Special signal conditioning circuits have been built to provide power, modulation, gain, and filtering of the sensor signals. With a sufficient number of proximity

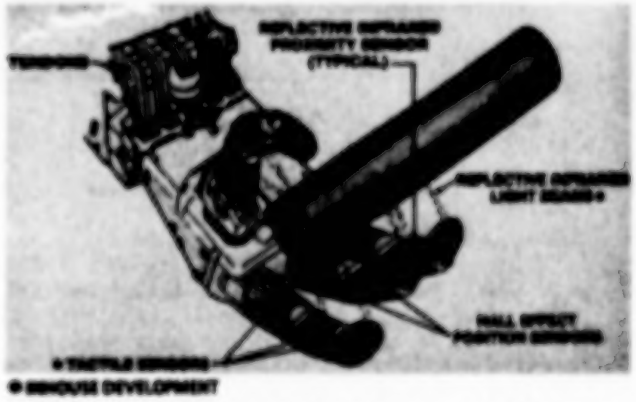


Figure 7.- The Utah/MIT dexterous robotic hand.



Figure 8.- Miniature IR proximity sensor.

sensors, a dexterous robotic hand would be capable of trapping an object by partially wrapping around it before coming into contact with it [8]. This will greatly increase the likelihood of success for the first grasp attempt. With these proximity sensors and the associated control system, the Utah/MIT hand is able to autonomously and securely grasp different shapes of objects and even catch objects tossed to it.

Control System

The EVAR Smart Hand control system development includes the design of servo control electronics using single-chip motor controllers, and the development of a VME bus based multi-processor control system using the Motorola 68020 32-bit microprocessors. The motor controller chip accepts encoder algorithm which resides on the chip itself. The execution of the control algorithm is invisible to the user, but the user has the ability to monitor the trajectory status and adjust the control parameters. Twelve motor controller chips fitted on a single PC expansion board that plugs into an IBM-AT motherboard have been used to control the Stanford/JPL hand. Since the trajectory of each motor is executed simultaneously by each motor controller chip independent of the host, the host central processing unit (CPU) is freed to execute other software routines. All high-level controls and some servo-level controls are accomplished by the VME bus based multi-processor control system using the 68020 microprocessors running the Software Components Group PSOS real-time operating system. A SUN 3/260 serves as the host for [illegible] control software development [9].

D. [illegible] an optimized EVAR Smart Hand design

Through the evaluations of the Utah/MIT hand, the Stanford/JPL hand, and the Direct Link Prehensor, the Direct Link Prehensor was found to have the most optimized finger arrangement, considering the trade-offs between complexity and functions. The anthropomorphic design of the Utah/MIT hand is desirable because it approximates an EVA crewmember's hand size and dexterity. But its remotizer and its large control electronics and actuator package make it enormously difficult to be packaged within the EVAR. The Stanford/JPL hand has fewer DOF than has the Utah/MIT hand, but its non-anthropomorphic design makes it difficult to share the same grasp interfaces as those used by an EVA crewmember. The Direct Link Prehensor has only 6 DOF in an anthropomorphic finger arrangement. This arrangement was found to be more adequate for grasping. Furthermore, the thumb is oriented approximately 120 degrees from the index finger. This configuration also allows some manipulation capability. The capability of the Direct Link Prehensor to grasp in weightless environment was validated on a KC-135 aircraft zero-gravity experiment [7]. Based on the evaluations, the Direct Link Prehensor has been selected as the baseline configuration for the Phase II EVAR Smart Hand.

With the EVAR Smart Hand baseline configuration selected, a motorized version of this hand was built by NASA/JSC and named the Jameson hand after the designer of the Direct Link Prehensor and the Jameson hand - Dr. John Jameson. The Jameson hand, as shown in Figure 9, has an integrated hand-wrist-forearm package that approximates the combined size of a human hand, wrist, and forearm. There are seven DC motors packaged in the forearm, with one motor per each DOF, plus one that controls the tendon tension.



Figure 9.- Jameson hand.

The wrist on the Jameson hand comes from a remote RM-10A arm that is used on the EVAR. Power is transmitted from the motors through a tendon-pulley system to each joint, much like the remotizer in the Utah/MIT hand. This tendon-pulley system allows the hand to move freely with the wrist. The encoders on each motor and the strain gauges in the hand provide position and force feedback. Proximity sensors have been installed on the Jameson hand to provide autonomous adaptive grasping capability. The Jameson hand control system consists of servo motor controllers and the VME 68020 CPU's. The partition of tasks for parallel processing and the selection of motor controller chips for the Jameson hand control system were derived from the control system evaluations. The fabrication of the Jameson hand has been completed, and the software and the control system for the Jameson hand have been developed in the CTSD EVA Robotics Laboratory. The Jameson hand will be integrated with the EVAR in Phase II for dynamic evaluations.

E. Integrate prototype hand with EVAR ground demonstration

The EVAR ground demonstration system contains two NASA/JSC developed hands: the Jameson hand on the right arm of the EVAR, and the CTSD II hand on the left arm of the EVAR. The Jameson hand control

system CPU's and motor controllers are located inside the robot body. The power semiconductors which deliver current to the motors are packaged inside the forearm. The signal conditioning circuit for the CTSD II hand is located in the forearm. The motor controllers that control the CTSD II hand are contained in a small box mounted inside the robot body. Once the hands are mounted on the EVAR, the arm-hand coordination algorithms and the dynamic interactions between the robot body and the arms and hands will be evaluated. The Grasp Region Analysis Software Package (GRASP), a non-model based grasp software package developed by NASA/JSC to determine the proper grasp location based on 3D laser images, will be used to position the arms and hands. Figure 10 shows the Phase II EVAR ground demonstration system with the Jameson hand and the CTSD II hand.

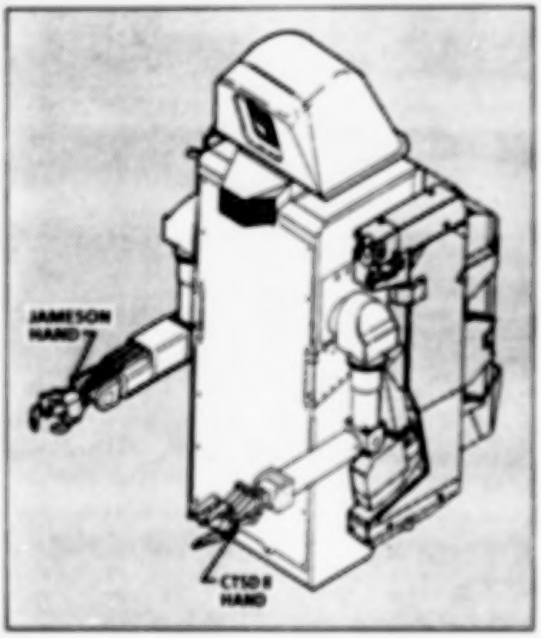


Figure 10.- EVAR Phase II hardware configuration.

FUTURE WORK

The development of EVAR Smart Hand is the first step in achieving the ultimate dexterous robotic hand capability. The next step is to develop autonomous manipulation capability. Pre-programmed robotic hand manipulations have been achieved through a teach-and-playback method. A more adaptive, real-time, intelligent, dexterous robotic hand manipulation capability will be pursued, using the Utah/MIT hand, the Stanford/JPL hand, and the Jameson hand as test beds. Dual-hand coordination will be evaluated by installing two dexterous hands on two robotic arms and selecting candidate tasks for evaluation. Tactile sensors will be evaluated and applied to the robotic hands to determine the contract locations and the directions of internal forces between the fingers and the object in order to provide adequate sensory feedback for intelligent robotic hand manipulation. Neural networks and other artificial intelligence software architectures will be explored for control and sensor fusion applications.

CONCLUSIONS

Several commercially available and in-house developed robotic hands have been evaluated for the development of the EVAR Smart Hand. The evaluations have shown that a dexterous robotic hand with 6 DOF in an anthropomorphic finger arrangement is sufficient for adaptive grasping. An autonomous, adaptive grasping concept was demonstrated on the Utah/MIT hand using position, force, and proximity sensors. The Jameson hand was developed with these features incorporated in a self-contained, hand-wrist-forearm package that approximates the combined size of a human hand, wrist, and forearm. The Jameson hand was designed to satisfy the objectives and the performance goals of the EVAR Smart Hand and will be integrated with the EVAR ground demonstration system in Phase II. Autonomous, robotic hand manipulation capability will be pursued in the follow-on efforts to develop a dexterous robotic hand system with capabilities approaching that of an EVA crewmember.

REFERENCES

1. EVA Retriever Program Plan, JSC-22144, Johnson Space Center, Houston, TX, May 1987.

2. EVA Retriever Phase I Final Report, JSC-23175, Johnson Space Center, Houston, TX, July 1988.

3. Reuter, G., Hess, D., et al, "An Intelligent Free-flying Robot," a paper presented at SPIE Symposium on Advances in Intelligent Robotic Systems, Space Station Automation IV, Cambridge, MA, Nov 6-11, 1988.

4. Jacobsen, S. C., et al, "Design of the Utah/MIT Dexterous Hand," IEEE International Conference of Robotics and Automation, San Francisco, CA, April 1986.

5. "The Force Sensing Resistor - A New Tool in Sensor Technology," a technical report by Interlink Electronics, Santa Barbara, CA, June 1986.

6. Bach-y-Rita, P., Crabb, T., et al, "Research Investigations in Sensory Augmentation Devices: Space Suit Glove Application," NAG9-171, a Final Report to NASA Johnson Space Center by University of Wisconsin and Astronautics Corporation of America, 1988.

7. Jameson, J., "Report on the Zero-G Grasp Experiment with the Direct Link Prehensor," NASA Johnson Space Center, March 15, 1988.

8. Wampler, C., "Multi-processor Control of a Telemanipulator with Optical Proximity Sensors," The International Journal of Robotics Research, Vol. 3, No. 1, Spring 1984.

9. Narasimhan, S., "Dexterous Robotic Hands: Kinematics and Control," Master of Science thesis in Electrical Engineering and Computer Science at MIT, January 1988.

KINEMATICALLY REDUNDANT ARM FORMULATIONS FOR COORDINATED MULTIPLE ARM IMPLEMENTATIONS

Robert W. Bailey and Leslie J. Quiocho
LinCom Corporation
1020 Bay Area Blvd, Suite 200 Houston, Tx 77058

Dr. Timothy F. Cleghorn
Mission Planning and Analysis Division - FM7
NASA/Lyndon B. Johnson Space Center Houston, Tx 77058

ABSTRACT

Although control laws for kinematically redundant robotic arms were presented as early as 1969 [1], redundant arms have only recently become recognized as viable solutions to limitations inherent to kinematically sufficient arms. The advantages of run-time control optimization and arm reconfiguration are becoming increasingly attractive as the complexity and criticality of robotic systems continues to progress. This paper presents a generalized control law for a spatial arm with 7 or more degrees of freedom (DOF) based on Whitney's resolved rate formulation [1]. Results from a simulation implementation utilizing this control law are presented. Furthermore, results from a two arm simulation are presented to demonstrate the coordinated control of multiple arms using this formulation.

INTRODUCTION

Within a resolved rate motion control scheme, the joint motors are run simultaneously to provide varying joint velocities consistent with constant commanded point of resolution (POR) velocities in Cartesian space. The fundamental relationship between joint or configuration space and task space velocities is the Jacobian matrix, which maps a linear transformation between the two spaces. For kinematically redundant arms, transforming task space commands (6 DOF) to joint space consisting of 7 or more DOFs requires resolving the redundancy, i.e., solving an underdetermined set of equations.

Two primary techniques have been proposed to resolve the redundancy as applied to robotic systems: the method of Lagrange multipliers and the generalized or pseudoinverse technique. Whitney [1,2] uses Lagrange multipliers with an optimality criterion to be satisfied during the motion of the manipulator. Liegeois [3] utilizes generalized inverse matrices (also referred to as pseudoinverse matrices) and adds to the solution a minimization vector representing the deviation from the mean positions of each of the joints. Klein and Huang [4] similarly incorporate a function minimizing excursion from the joint center positions. Bourgeois [5] implements Whitney's algorithm with a weighting matrix to keep joints from approaching motion limits during a trajectory. In addition to these efforts, other optimality algorithms have been proposed to provide avoidance with obstacles [6] and to minimize torque loading on the joints in a least squares sense [7].

Despite these efforts, few results have been presented demonstrating the physical performance characteristics of the control laws [3,5]. Furthermore, literature on spatial formulations and simulation implementations of kinematically redundant arms is scarce (although numerous commercial vendors are marketing hardware with spatial 7-DOF control laws [8]). As a result, the focus of this effort is on 1) the spatial implementation of a generalized resolved rate law for a 7 or more DOF arm based on Whitney's original formulation, and 2) the physical performance characteristics of the control law.

FORMULATION

Equation (1) defines the robotic arm POR velocities as functions of the joint velocities. This relation may be derived using methods of classical mechanics [9] based on the parameter definitions depicted in Figure 1†.

$$\begin{bmatrix} \Omega \\ V \end{bmatrix} = [\dot{x}] = [J]\left[\dot{\theta}\right] \qquad (1)$$

where J, the Jacobian matrix, is defined as

$$J = \begin{bmatrix} U_1 & U_2 & \cdots & U_n \\ U_1 \times R_1 & U_2 \times R_2 & \cdots & U_n \times R_n \end{bmatrix}.$$

The Jacobian J is an $m \times n$ matrix, where m represents the dimension of the task space and n the dimension of the joint space. In the case of a kinematically redundant system, $n > m$ and the system of equations is underdetermined, i.e., J is nonsquare and the inverse of J is undefined. Hence, additional criteria need to be introduced to produce a unique solution.

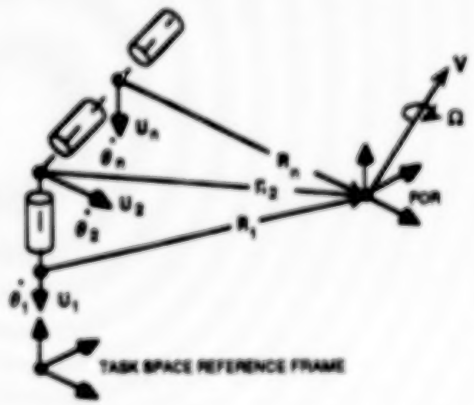


Figure 1 - N DOF Configuration Space Parameters

†Manipulator joint rates $\dot{\theta}$ previously defined as $\dot{\gamma}$ in [9].

The method of Lagrange multipliers [10] can be used to optimize an objective or cost function subject to a given constraint. For the problem at hand, the method can be used to "square" the system of equations, thereby producing an invertible solution to Equation (1). The problem can be formulated in the following manner: suppose f and g are vector functions of some joint angle vector $\dot{\theta}$. If $f(\dot{\theta})$ has a local extrema (minimum or maximum) at $\dot{\theta}_e$ subject to the constraint $g(\dot{\theta})$, there must exist constants (Lagrange multipliers), Λ, such that the following expression is satisfied,

$$\nabla f(\dot{\theta}_e) = \Lambda \nabla g(\dot{\theta}_e). \qquad (2)$$

Whitney [1,2] introduces the following objective function in matrix form,

$$f(\dot{\theta}) = 1/2\ [\dot{\theta}]^T A [\dot{\theta}], \qquad (3)$$

where A is an $n \times n$ diagonal matrix of weighting terms. The constraint function, from Equation (1), is given by

$$g(\dot{\theta}) = [\dot{x}] = [J][\dot{\theta}]. \qquad (4)$$

Applying Equation (2) to Equations (3) and (4) yields the following expression,

$$[A][\dot{\theta}] = [J]^T[\Lambda] \qquad (5)$$

which can be rewritten as

$$[\dot{\theta}] = [A]^{-1}[J]^T[\Lambda]. \qquad (6)$$

After applying the method of Lagrange multipliers, matrix algebra can now be used to form the inverse solution to Equation (1). Substituting Equation (6) into Equation (4), and solving for Λ results in

$$[\Lambda] = \{\ [J][A]^{-1}[J]^T\ \}^{-1}\ [\dot{x}] \qquad (7)$$

Finally, the expression for Λ in Equation (7) can be substituted into Equation (6) resulting in the following resolved rate control law,

$$[\dot{\theta}] = [A]^{-1}[J]^T\ \{\ [J][A]^{-1}[J]^T\ \}^{-1}\ [\dot{x}] \qquad (8)$$

Alternatively, this solution could have been produced using a pseudoinverse method [1].

There seems to be some confusion in the literature as to the physical nature of the optimization function introduced in Equation (3). Whitney refers to this criterion as the "approximate instantaneous weighted system kinetic energy." Since the time of Whitney's publication, this statement has received several different interpretations. For example, Klein and Huang [4] state that "... since energy consumption can be related to the norm of joint velocities, and since the pseudoinverse finds the minimum norm solution, instantaneous power is minimized." Baker and Wampler [11] propose finding "... the joint speeds $\dot{\theta}_e$ which minimize some norm of $\dot{\theta}$, such as $\dot{\theta}^T\dot{\theta}$ or kinetic energy." The actual physical interpretation of this minimization criterion is illustrated in Figure 2. If the A matrix contains the moments of inertia of each of the arm segments about their axis of rotation, then Equation (3) represents the sum of the rotational kinetic energies of each of the segments, as if each segment was a 1 DOF system. In essence, Equation (3) seems to provide a convenient means to invert the equation set of Equation (1), rather than providing an actual physical criterion to minimize or maximize.



Figure 2 - Physical Interpretation of Objective Function

The vectors Ω_c and V_c, Equations (9) and (10) respectively, correspond to the commanded POR rotational and translational velocities. These parameters are graphically defined in Figure 3.

$$\Omega_c = u^T[T_o] \cdot \Omega_{lim} \qquad (9)$$

$$V_c = \text{unit}\ ([R_f - R_o]) \cdot V_{lim} \qquad (10)$$

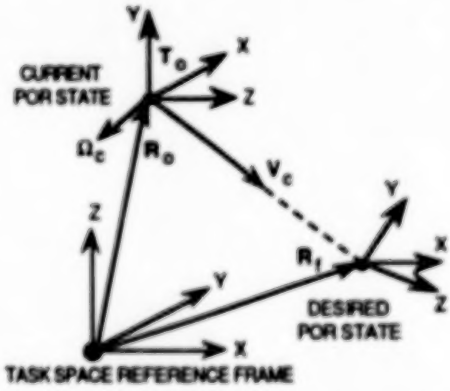


Figure 3 - POR Rotational & Translational Velocity Commands

Equations (9) and (10) produce linear, goal oriented command inputs to the control law [9]. In other words, for an ideal system, Equations (9) and (10) will produce linear paths to the desired POR orientation and position, both in a rotational and translational sense. For an imperfect system (sensor corruption, motor lag, etc.), actual POR paths will have a tendency to "spiral in" to the desired orientation and position, always moving towards the goal point regardless of the actual path taken.

SIMULATION IMPLEMENTATION

The control law of Equation (8) is implemented within the Robotics Software Testbed (RST) architecture developed by the Mission Planning and Analysis Division at the Lyndon B. Johnson Space Center. The RST provides a discrete time cyclic executive with the capability of integrating both hardware and software components into a simulation for both ground based and orbital applications. The application discussed here is generalized for seven or more DOF and data driven so that different arm configurations may be tested without code recompilation. A high level schematic flow diagram of the single arm simulation is presented in Figure 4.

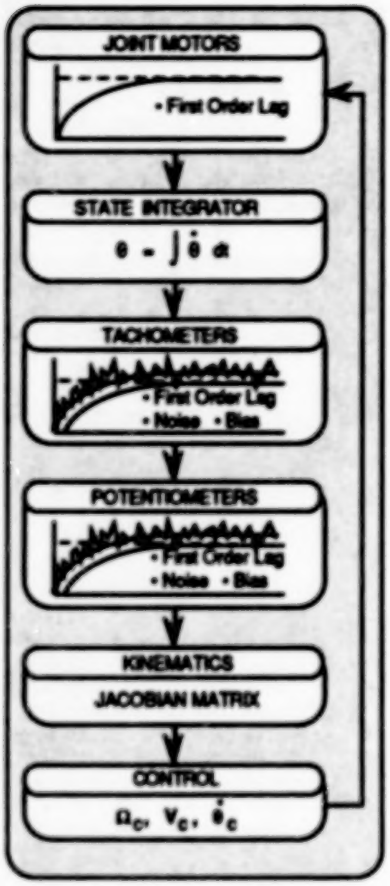


Figure 4 - Kinematic Arm Simulation Schematic Flow Diagram

The Robotics Research Corporation's (RRC) Model K-1607™ seven DOF arm was selected as the test configuration for initial analysis [8]. The arm has a reach of approximately five feet and a R-P-R-P-R-P-R joint configuration, as depicted in Figure 5. Numerous simulation runs were performed to help analyze the response of the control law. Specifically, five runs were chosen which best demonstrate the effects of the ***A*** matrix weighting on control performance.



Figure 5 - Robotics Research Corporation Model K-1607™

The test case scenario used for each run consists of a single POR move from an initial to a final position and orientation in the task space. Each of the five simulations are identical upon initialization except for the weighting of the ***A*** matrix. All runs are executed with a 12.5 Hz control loop cycle. The specific differences between the ***A*** weighting for each of the runs are as follows:

Run 1 – constant unity weighting over duration of the run.
Run 2 – constant weighting over duration of run with joint 6 at two orders of magnitude above all other joints.
Run 3 – constant weighting over duration of run with joints 2 and 6 at two orders of magnitude above all other joints.
Run 4 – varied weighting over duration of run based upon actual joint distance from the joint median angle.
Run 5 – a dual arm simulation with the characteristics of Run 1 for both arms.

The weighting algorithm used for Run 4 performs the following steps:

1) find the minimum and maximum percentage deviations from the median angles among the joints (percentage deviation is the actual angular deviation over the actual angular displacement to the joint limit for the joint).

2) scale the ***A*** component corresponding to the joint with the minimum percentage of deviation to 1.0. Scale the ***A*** component corresponding to the joint with the maximum percentage of deviation to 100.0. Scale all other ***A*** components proportionally based on their respective joint percentage deviations.

Run 5 is presented to show the control law's ability to support multiple arm coordination. Cleghorn and Bailey [9] discuss the coordination of 6 DOF robotic arms utilizing the same linear, goal oriented command generation scheme of Equations (9) and (10). Run 5 consists of a concurrent implementation of two identical RRC arm models as used in Runs 1 through 4. The arms are configured side by side with the POR of arm 1 placed at the tip of the last joint segment (as in Runs 1 through 4), and the POR of arm 2 placed 0.2 feet off the end of the last joint segment and 1.0 feet to the side; this places the POR between the two arms. The arms are initialized with their PORs at the same location in the task space, and then given identical POR maneuver commands (the same maneuver executed for Runs 1 through 4).

RESULTS

Several interesting results are found common to all test runs. First, there is no appreciable change in the POR trajectory between the runs. This stability demonstrates the robustness of the control algorithm while performing a maneuver, given a wide variety of ***A*** weighting values and joint solutions. Although the POR trajectories were uniform over the five runs, the joint angle and joint rate histories differed considerably.

Second, the ***A*** matrix values are important in a relative rather than an absolute sense. For example, Run 1 was executed with 1) all ***A*** values equal to 1.0, and 2) all A values equal to 10000.0. No POR or joint differences were observed between these two cases. Also, Run 2 was executed with 1) the ***A*** values for joint 6 set at 100.0 and the remaining values set to 1.0; and 2) the ***A*** values for joint 6 set at 100000.0 and the remaining values set to 1000.0. For both runs, the ratio of the largest ***A*** value to the smallest ***A*** value was 100 (two orders of magnitude). Once again there was no noticeable POR or joint

differences were observed between the two runs. This shows the importance of the relative magnitudes of the *A* values.

Third, as shown by Bourgeois [5], increasing a single value of the *A* matrix relative to the other values has the effect of limiting the motion of that respective joint. In general, as the ratio of the largest to smallest *A* values falls below 100, the joint histories approach that of the all unity case (Run 1). As the ratio increases above 100, the weighting tends to increase the motion of all the low *A* value joints rather than decreasing the motion of all the high *A* value joints.

While all of the five runs exhibit common characteristics, there are some specific results of the individual runs worth mentioning. In Figure 6 notice that the acceleration, maneuver velocity, and deceleration regions of the POR velocity profiles are nearly linear. These velocity profiles result in linear POR position histories. Although the rotational velocity vector components are approximately linear, the orientation histories exhibit a slight curvature throughout the duration of the run. This can be attributed to the fact the velocity histories are presented as a rotational velocity vector in the task space frame, whereas the orientation histories are presented as pitch-yaw-roll Euler angle sequences. As discussed by Cleghorn and Bailey [9], this linear response can be used to an advantage in multiple arms systems. If each manipulator of a multiple arm system can be expected to follow a linear trajectory at a given speed, then the identical POR velocity commands given to each of the independently controlled arms should trace the same linear POR path. Further dual arm results are discussed for Run 5.

Figures 7 and 8 present joint angle and joint rate histories for each of Runs 1 through 4, where Run 1 represents the nominal case (unity weighting). Run 2 demonstrates the powerful ability of the *A* matrix weighting to limit the motion of a single joint; in this case joint 6. However, notice that the restricted motion of joint 6 induces more motion in the most of the remaining joints.

Run 3 is similar to Run 2, but with two joints weighted to limit their motion (joints 2 and 6). In comparison to the unity case (Run 1), motion for joints 2 and 6 is inhibited while motion for the remaining joints increases. In general, as more joints are weighted to limit their motion, their respective joint histories approach that of Run 1. As was previously mentioned, the relative magnitudes of the weighting elements are important and not their absolute magnitudes.

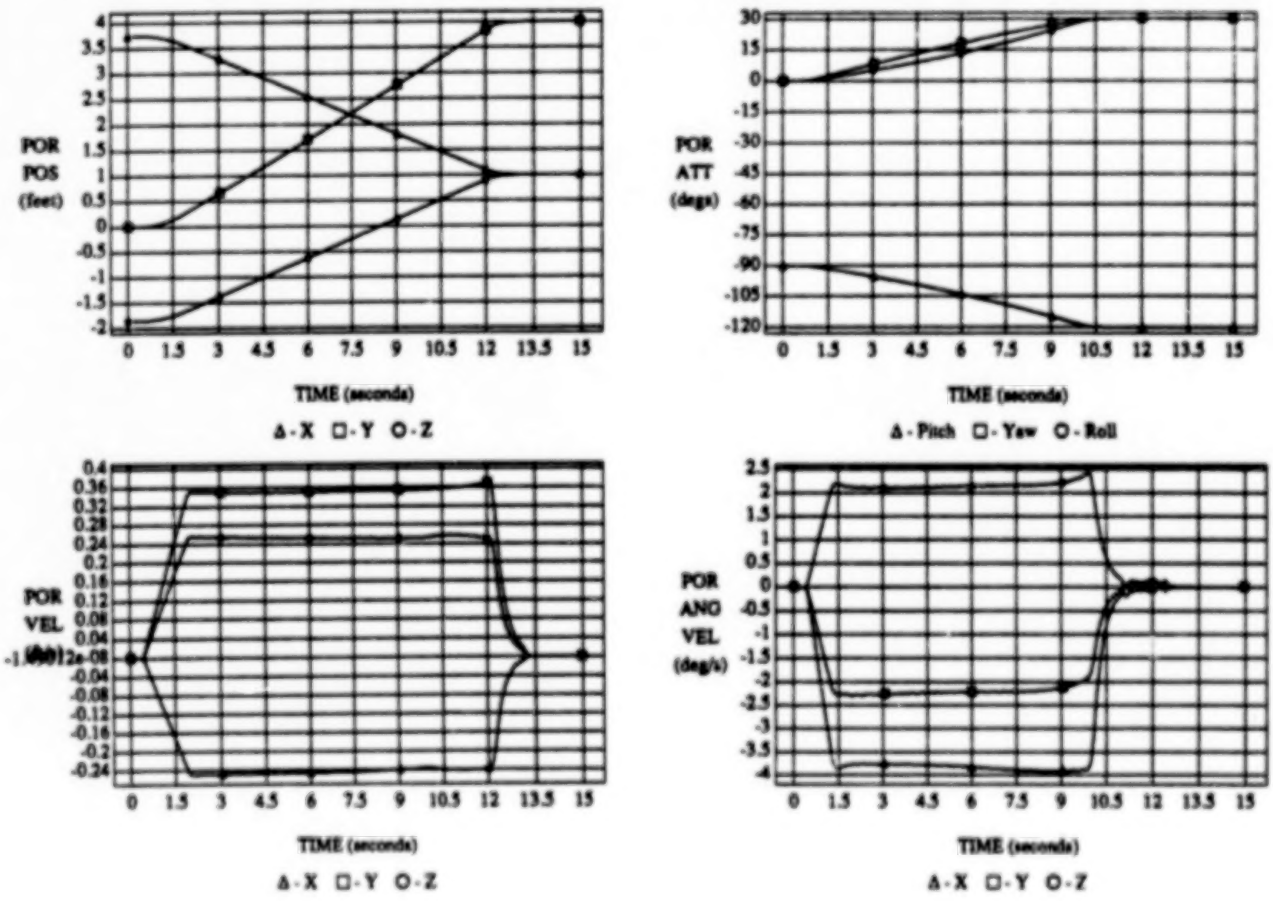
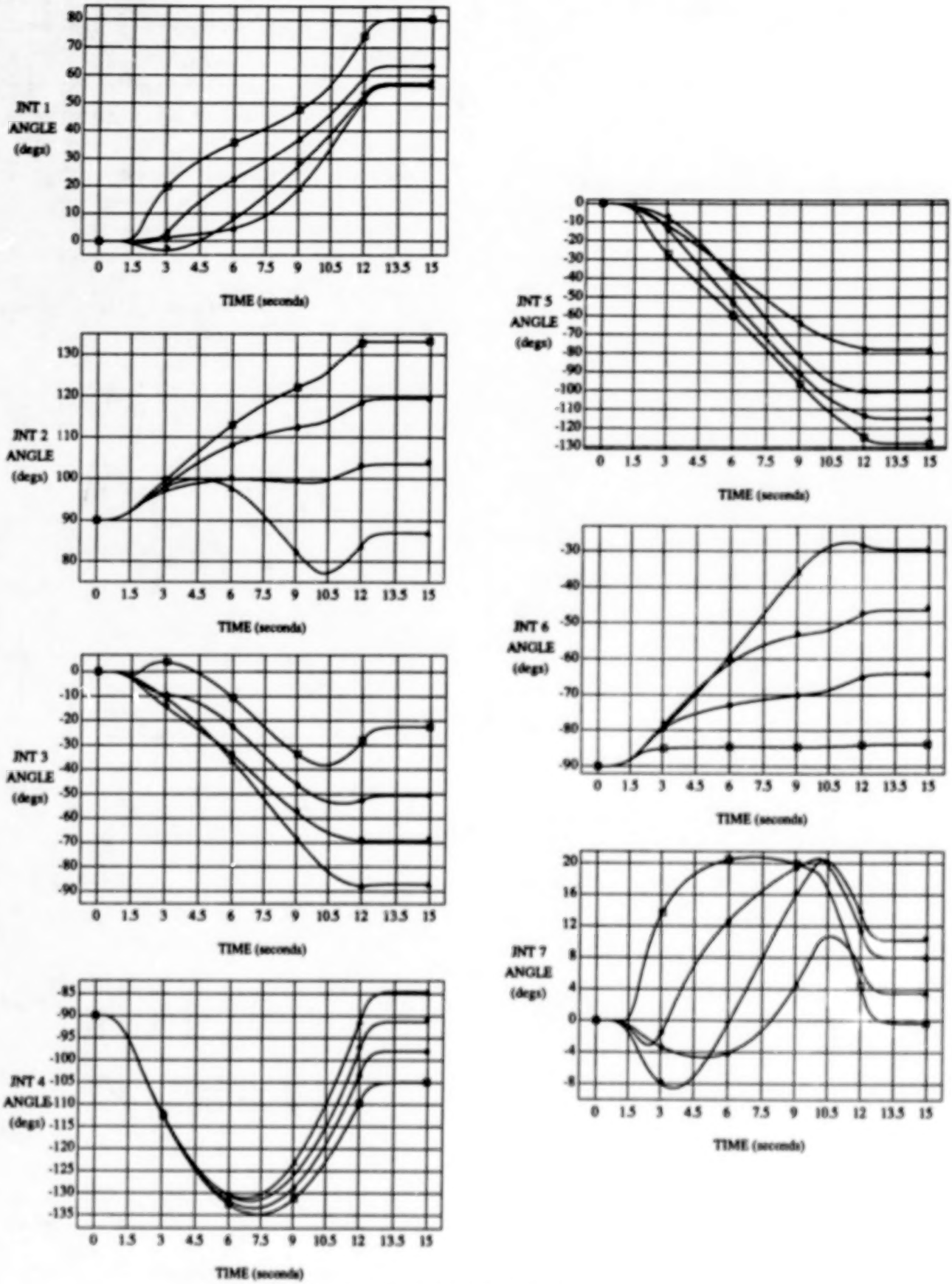


Figure 6 - POR Position and Velocity Histories for Run 1



Δ - Run 1 □ - Run 2 O - Run 3 • - Run 4

Figure 7 - Joint Angle Histories For Runs 1 Through 4

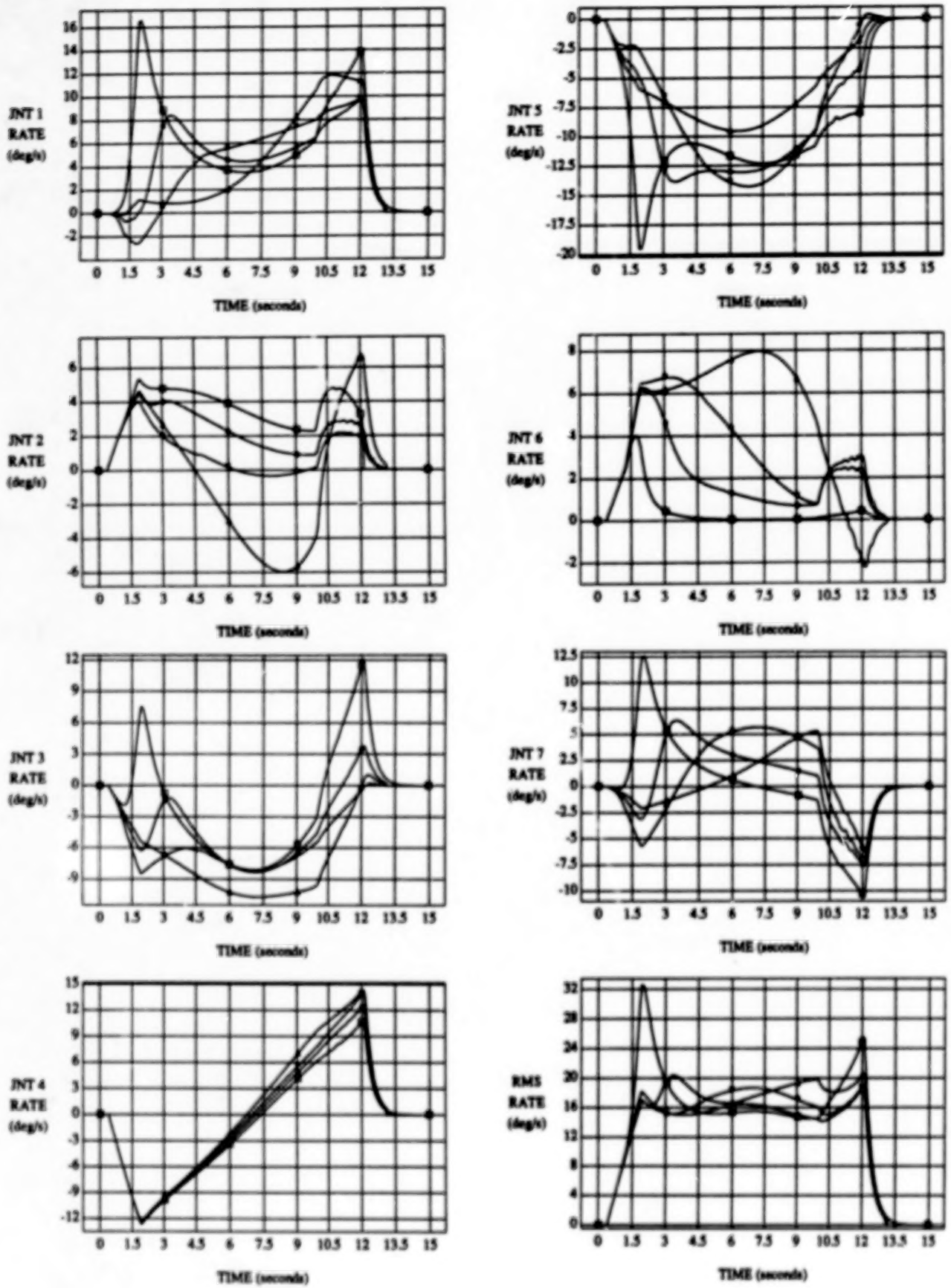


Figure 8 - Joint Rate Histories For Runs 1 Through 4

Run 4 differs from the previous three runs in that the weighting is performed dynamically rather than remaining constant. As the joint travels farther from the median position, the weighting is scaled to impede further motion away from the center position, which is not to be confused with attracting the joint to the center position. The effects are reflected in Figures 7 and 8. Since all joints are being weighted simultaneously, not all joints can be expected to fulfill the criteria of remaining close to the center position. One would expect the three joints farthest away from their center angles to exhibit the slowest motion. Likewise, the three joints closest to their center positions should exhibit increased motion and the remaining joint should stay relatively unchanged from the nominal (Run 1). In general, Run 4 demonstrates these results; as the data presented in Table 1 confirms. A plus in the PERFORM row of Table 1 indicates motion was slowed compared to Run 1; a zero indicates relatively no change in motion; and a minus indicates an increase in motion.

Table 1 - Configuration Data Pertinent to Run 4

JOINT	1	2	3	4	5	6	7
CENTER ANGLE	0.0	45.0	0.0	-90.0	0.0	-90.0	0.0
LOW HARDSTOP	-360.0	-45.0	-360.0	-180.0	-360.0	-180.0	-1080.0
HIGH HARDSTOP	360.0	135.0	360.0	0.0	360.0	0.0	1080.0
% DEVIATION @ 15s	16	66	19	2	28	49	1
+/0/- PERFORM	0	+	+	+	-	+	-

Run 5 is included to show the control law's ability to enable multiple arm control based on the methodologies proposed by Cleghorn and Bailey [9]. Figure 9 shows the root mean square (RMS) POR position and attitude error between arms 1 and 2 over the duration of the maneuver. Although the maximum errors of one third of an inch and half of a degree may appear slightly large, the proposed goal oriented resolved rate control law is self correcting. For the kinematic simulation of Run 5, the PORs of the two arms are not constrained to be coincident. However, for an actual hardware implementation, the PORs of the two arms would be constrained to be coincident while grasping the same object. Since the control law is resolved rate rather than resolved motion, rate sensory feedback will reduce the positional errors between the two arms, thereby reducing the opposing forces and torques. This error reduction is the self-correcting effect mentioned earlier.

As an aside, notice that the joint and joint rate time histories for joint 4 remain relatively unchanged. This can be explained by the nature of the maneuver, which is primarily translational. For this particular trajectory, joint 4 pitch is critical to attain desired POR translation.

CONCLUSIONS

A generalized control law for a spatial arm with 7 or more degrees of freedom (DOF) has been presented. This control law was derived using Lagrange multipliers to optimize a function consisting of squared joint rate terms. This function can be physically interpreted as the summed individual link rotational kinetic energies when the A weighting values are set to the link inertias.

Results from a simulation implementation utilizing this control law were presented to clarify the effects of the associated weighting matrix. Despite the variance in joint angle solutions and corresponding rates, the linear response of POR in task space is maintained. This ability is ideal for multiple arm control implementations utilizing independent control laws for each arm.

The relative rather than absolute magnitudes of the A weighting matrix are important. Specifically, preliminary results indicate the "magic 100" ratio to be a good engineering solution. Increasing a single value of the weighting matrix tends to limit motion of the corresponding joint; however, this effect is reduced when applied to a higher number of joints.

Although the current analysis is based upon Whitney's formulation, other formulations might be more applicable to minimizing practical kinematic quantities, such as joint travel. Implementation of additional formulations within the RST will be an immediate subject of future research. Integrating and analyzing collision avoidance and path planning algorithms for 7 or more DOF robotic arms is also planned.

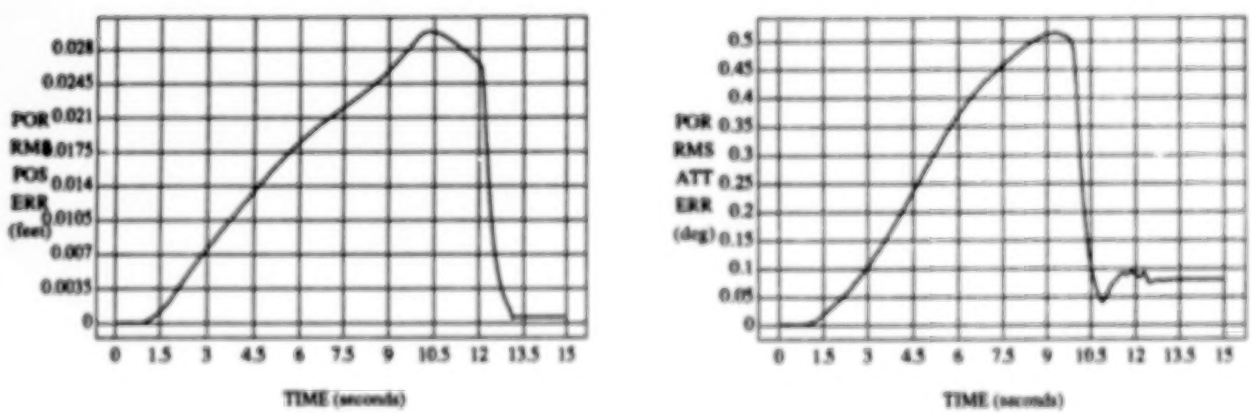


Figure 9 - RMS POR Errors Between Arms 1 and 2 for Run 5

REFERENCES

[1] Whitney, D. E., "Resolved Motion Rate Control of Manipulators and Human Prostheses," *IEEE Transactions on Man-Machine Systems*, Vol. MMS-10, No. 2, 1969, pp. 47 - 53.

[2] Whitney, D. E., "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 122, 1972, pp. 303 - 309.

[3] Liegeois, A., "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms, *IEEE Transactions on Systems, Man, and Cybernetics*, n Vol. SMC-7, No. 12, 1977, pp. 868 - 871.

[4] Klein, C. A., and Huang, C., "Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-13, No. 3, 1983, pp. 245 - 250.

[5] Bourgeois, B. J., "An Optimal Resolved Rate Law for Kinematically Redundant Manipulators," *First Annual Workshop on Space Operations Automation and Robotics (SOAR 87)*, 1987, pp. 457 - 464.

[6] Maciejewski, A. A., and Klein, C. A., "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," *International Journal of Robotics Research*, Vol. 4, No. 3, 1985, pp. 109 - 117.

[7] Suh, K. C., and Hollerbach, J. M., "Local versus Global Torque Optimization in Redundant Manipulators," *IEEE Conference on Robotics and Automation*, Vol. 1, 1987, pp. 619 - 624.

[8] Karlen, J. P., Thompson, J. M., and Farrell, J. D., "Design and Control of Modular, Kinematically-Redundant Manipulators," *Second AIAA/NASA/USAF Symposium on Automation, Robotics, and Advanced Computing for the National Space Program*, March 1987.

[9] Cleghorn, T. F., and Bailey, R. W., "Task Space Coordination of Multiple Robotic Arms Manipulating the same Payload in an Orbital Environment," *Satellite Servicing Workshop IV*, June 1989, proceedings to be published.

[10] Mardson, J., and Tromba, A., Vector Calculus, W. H. Freeman and Company, San Fransisco, 1981.

[11] Baker, D. R., and Wampler, C. W., "Some Facts Concerning the Inverse Kinematics of Redundant Manipulators," *IEEE Conference on Robotics and Automation*, Vol. 1, 1987, pp. 604 - 609.

Numerical Approach of Collision Avoidance and Optimal Control on Robotic Manipulators

Jyhshing Jack Wang, Automation Specialist, The M. W. Kellogg Company

The author owes help from: Jyhpyng Wang, Research Associate, M.I.T.

Abstract

Collision-free optimal motion and trajectory planning for robotic manipulators are solved by a method of sequential gradient restoration algorithm. Numerical examples of a two degree-of-freedom (DOF) robotic manipulator are demonstrated to show the excellency of the optimization technique and obstacle avoidance scheme. The obstacle is put on the midway, or even further inward on purpose, of the previous no-obstacle optimal trajectory. For the minimum-time purpose, the trajectory grazes by the obstacle and the minimum-time motion successfully avoids the obstacle. The minimum-time is longer for the obstacle avoidance cases than the one without obstacle. The obstacles avoidance scheme can deal with multiple obstacles in any ellipsoid forms by using artificial potential fields as penalty functions via distance functions. The method is promising in solving collision-free optimal control problems for robotics and can be applied to any DOF robotic manipulators with any performance indices and mobile robots as well. Since this method generates optimum solution based on Pontryagin Extremum Principle, rather than based on assumptions, the results provide a benchmark against which any optimization techniques can be measured.

Key words Bang-bang control, optimal control, Cartesian space, joint space, robotic manipulators, degree-of-freedom.

1. Introduction

The problem of increasing productivity, automated manufacturing, and performing complex tasks in hazardous or remote environments can be solved by robotic systems. Such systems have been applied to a wide variety of industries which includes spray painting, welding, assembling, material handling, highly risky work and remote control jobs. As pointed out by Holcomb and Montemerlo [1] and Lerner [2], remote control robotic systems will be developed in the future space stations. Also as well-known, with the demand of increasing productivity and industrial automation, the problem of controlling the robotic manipulators has received a great deal of interest in the field of automated manufacturing.

1.1 Research Objectives

One of the focal points in robot design lies in the computation and control of the motion of the manipulator. In order to make sure that the manipulator is able to execute a special task most efficiently for human beings, it is important that the manipulator performs from initial states to designated final states in an optimal way under collision avoidance concern. Control on robotics can be separated into two major categories: 1) trajectory planning, 2) trajectory tracking. Various optimal controllers need to be devised in the trajectory tracking problems which are not the subjects in this article. Trajectory planning is not only the determination of the path of the end effector. Trajectory planning generates a specified motion of time history from initial states to final states. Motion planning does not necessarily require optimization techniques but extra excursion of the robot is just not cost-efficient and can cause more potential collision problems. Obviously, the minimum-time trajectory is of particular interest since the productivity in automation is maximized. Various performance goals, for example: distance, energy or time-energy combination, are also applicable. Various concepts for the study of optimal control of robotic manipulators have been studied for this purpose.

1.2 Previous Work

One of the pioneered work is done by Kahn and Roth [3]. The highly nonlinear manipulator dynamical equations of motion are linearized, an approximate bang-bang solution has been developed to the suboptimal feedback control problem.

Gilbert and Johnson [4] have developed a path planning scheme in which the obstacles are avoided via an infinite penalty function generated from distance function. In their study, the nonlinear dynamic equations are approximated by linear subspace functions which are chosen as piecewise polynomial splines. In their examples, distance constraints are violated when spline knot interval sections equal to one; the payload object is made strictly convex by approximating its boundary by arcs of certain curvature; obstacles are assumed to be convex sets; the complex distance finding minimization problem within the optimal control problem is not fully described. In the optimization technique, more than one optima can be drawn at the same case.

Based on Pontryagin extremum principle, the time-optimal motions of various types of robotic manipulators have been investigated by Geering, Guzzella, Hepner and Onder [5] as classified by cylindrical, spherical robots, and a robot with horizontal articulated arm with two links. In the analysis of the time-optimal control problem, the bang-bang control solution satisfies the Pontryagin extremum principle and the study has been made for unconstrained trajectories. In their examples, two links intercross each other in the planar two-link manipulator.

Due to the difficulty of highly nonlinear robot mathematical model, a near-optimal control algorithm based on Pontryagin extremum principle and Riccati formulation has been presented by Kim, Jamshidi and Shahinpoor [6]. The algorithm reduces the original nonlinear equation set into a linear one by a parameter sensitivity method and P-D controller is used to solve the linearized model.

Ozaki and Mohri [7] has developed the study of collision-free joint trajectories along a given path with some physical constraints such as manipulator dynamics, obstacles avoidance, joint velocities and input torques by formulating artificial potentials into the planning problem for constraints using linear programming algorithm to minimize the error between present and desired trajectory, in which, the nonsmooth time functions were approximated by cubic spline functions.

The technique of dynamic programming has also been a popular solution method to many investigators in the field of robotic manipulators research. Based on dynamic programming, Vukobratovic and Kircanski [8] have determined the energy-optimal velocity distribution of the manipulator end-effector for a prescribed path in the workspace subject to the forces/torques constraints. The given traveling time needs to be discretized in their study.

Singh and Lee [9] have formulated and solved the optimal trajectory planning as an optimal control problem by a path parameterized method of dynamic programming under the constraints of the joint forces/torques and velocities. Bang bang control has been generated for minimum time problems without obstacles avoidance concern.

In order to implement dynamic programming approach, Shin and McKay [10] have studied trajectory planning of robotic manipulators using parametric function and its derivative to reduce dimensions in state space which finds the positions, velocities, accelerations, and torques of the problem by minimizing the cost of the parameter of moving a robotic manipulator along a specified geometric path subject to input torque/force constraints without obstacle avoidance concern. Along a pre-selected geometric path, for quadratic velocity bounds, and piecewise analytic geometric path constraints, the minimum-time control problem has been studied by Shin and McKay [11] with the phase-plane techniques in Cartesian space which has to be converted into joint space by interpolation. Under the assumption that the path is given as parameterized curve, they have also determined a near-minimum time geometric path for the study described above which minimizes approximate lower traversal time bounds using maximum velocity bounds [12]. Their techniques are limited by parameterization.

Bobrow, Dubowsky, and Gibson [13] have studied the problem of minimum-time trajectories along arbitrary pre-planned spatial paths by a special technique in which the actuator torque bounds are assumed to be functions of the robot's current position and velocity. This technique cannot handle the case when the feasible regions in the phase plane are not simply connected. The idea of the time-optimal solution is based on choosing the maximum acceleration/deceleration to make velocity as large as possible at every point without violating constraints. The difficulty is finding multiple switching points for time-optimal problems. Dubowsky, Norris and Shiller [14] have devised a time optimal trajectory planning scheme with obstacle avoidance consideration via a CAD approach in which the minimum distance to obstacles is found from software OPTARM II by a table of various geometric shape. The penalty function for obstacle avoidance needs to have a characteristic of more effective weighing and

dramatic steepness. The technique cannot be easily extended to solving optimal trajectory planning for other performance indices and constraints.

Based on the same assumption, Rajan [15] has devised a parameterized path method for the minimum-time problem in which the cubic spline paths are parameterized and optimized locally by an iterative scheme. The optimization procedure stops until the minimum-time path comes close enough to the previous path while using Bobrow's algorithm for inner minimization and varying the path for outer minimization. The algorithm cannot be effectively applied to the planar movements of a manipulator with obstacles in the workspace. The weak points of the algorithm are on the premises that the minimum time path is smooth and a smooth curve is well approximated by splines.

Saha and Hollerbach [16] have devised a method based on state-space search tree representing all possible solutions, and searching for the best one by using a Symbolics Lisp Machine for time-minimum criterion. The algorithm is a logical approach but not a mathematical approach which is not suitable for routine off-line trajectory planning due to the complexity of computation.

Lah and Lin [17] have devised a kinematical approach which assumes the path consists of a sequence of Cartesian straight line segments and constant limits on Cartesian velocity and acceleration are known a priori without considering the dynamics of the arm.

Weinrob and Bryson [18] have presented the Adjustable Control-Variation Weight (ACW) algorithm for the minimum-time control of a two-link robotic arm through choosing controls subject to the actuator constraints. In their examples, the two links of the planar manipulator intersects each other. Meier and Bryson [19] have developed an algorithm for solutions of time-optimal control problem of a two-link planar manipulator which contains solutions for two-point boundary value problem of constrainted motion between two endpoints.

Zhang and Wang [20] have investigated a collision-free time-optimal control problem of a two-link planar robotic manipulator by applying the method of global linearization transformation in joint space configuration. As a result, the nonlinear equations of motion are transformed into an equivalent linear model and an approximate explicit expression has been obtained for the case of minimum-time control of a two-link planar robotic manipulator with two-dimensional planar geometrical obstacle avoidance. In their example, radius of the circle obstacle is not shown.

Bobrow [21] has continued the study of optimal path planning using minimum-time criterion with obstacles avoidance consideration in which the actuator torque bounds are assumed to be functions of the robot's current position and velocity, where the Cartesian path of the end-effector is represented with uniform cubic B-spline polynomials. The obstacle avoidance is enforced by ensuring the distance between the end-effector and the obstacle which was evaluated by stepping small increments of the path parameter.

Wang [22] has devised the numerically approach of using sequential gradient restoration algorithm to solve Bolza classical optimal control problem on robotics without linearization or parameterization, including the analytical time-optimal solutions of a two-link manipulator and/or actuator constrainted cases, in which the implementation can be extended into obstacle avoidance consideration.

1.3 Overview

We can see that numerous attempts have been made to find collision-free optimal motion of a robotic manipulator without great success. All of the aforementioned investigations are limited in one way or another.

Collision-free optimal control problems for robotic manipulators are difficult due to the two-point boundary-value problem which involves, in addition to the optimality conditions, the kinematical and highly nonlinear dynamical equations of the system, the obstacle constraints, the limits imposed on controls, and the satisfaction of terminal conditions. Generally speaking, analytical solutions for classical optimal control problems with equality and/or inequality constraints are not possible. Therefore, numerical method is resolved. Numerical methods and computer routines are available nowadays ranging from simple integration to TPBVP and optimization at a low price [23].

To solve constrainted optimal control problems, a restoration phase is needed at the end of the gradient phase [24]. The collision-free motion planning problems of robotics can be formulated as a classical optimal control problem and solved by sequential gradient restoration algorithm [25]. Collision can be avoided by continuously controlling the closest point on the arm to the obstacles using virtual potential fields as penalty functions via distance functions [26].

1.4 Present Modeling

As pointed out in recent research, owing to the difficulty of solving TPBVP and highly nonlinear dynamic equations, the classical optimal control problem is mostly approached by approximation (linearization, parameterization, modification) which more or less replaces the original optimal control problem into the assumed one. As in those study where the nonlinear dynamic equations or the two-point boundary value problems are linearized or parameterized, the solutions generated based on those assumptions are not necessarily good approximations to the original ones. The intention of this research is to present a numerical approach for determining the collision-free optimum motion of robotic manipulators, a method to solve classical optimal control problem without any modification, linearization or simplification. Solutions including robot positions, velocities, accelerations and force/torque in both Cartesian space and joint space which satisfies the Pontryagin extremum principle are obtained by solving the manipulator kinematical and dynamical equations with optimality conditions. For given initial and final conditions, under the physical conditions imposed on control in joint space and obstacles constraints, the continuous time-history of the positions, velocities, accelerations, torques/forces and the optimal collision-free motion of a robotic manipulator in minimum time are determined.

Applications of sequential gradient restoration algorithm occur in various branches of science and engineering. With particular regards to aerospace engineering, various problems of coplanar and noncoplanar, orbital and suborbital space flight [27, 28, 29] and atmospheric flight in a windshear [30, 31, 32] have been solved by the sequential gradient restoration algorithm. Also, the same technique has been successfully employed in the thermofluid science [33, 34]. In general, sequential gradient restoration algorithm has proven to be a very promising algorithm in solving engineering optimal control problems [35, 36].

1.5 Advantages over Existing Techniques

To solve collision-free optimal control problems on robotics with constraints, we need a numerical method which has the following advantages:

1) able to solve TPBVP which is essentially the core of the problem [6, 8, 10, 11, 12, 13, 14, 15, 21]: In fact, TPBVP can be solved by shooting method and relaxation method or method of particular solutions. TPBVP is involved in the first-order exact optimality conditions derived from calculus of variation.

2) able to solve highly nonlinear dynamic equations without linearization, parameterization or simplification [3, 4, 6, 7, 10, 11, 12, 17, 20, 21]: Any modification by linearization or simplification directly or indirectly replaces the original problem. The drastic approximation leads to significant error and unsatisfactory, unknown effect to the optima and the obstacle avoidance. For example, a collision-free optimal solution can be declared only when there is not another more optimal solution.

3) able to solve any robotics formulation regardless number of joints or DOF [15, 17, 18, 19, 20]: A technical approach should not be limited by the number of joints or DOF of robotics. Any dynamic systems can be formulated from state functions point of view and solved as control systems regardless number of dimensions.

4) able to avoid the obstacles toward optimization direction without any unnecessary excursion [5, 7, 8, 9, 14, 15, 20, 21]: Collision avoidance should be achieved in a most efficient way, in terms, an optimal way, without requiring extra journey of the robot arm. The weighting effect and clearance between trajectory and obstacle should be specified by only one parameter.

5) able to solve general constraints of robot motion planning: On trajectory planning, we have state constraints, control constraints, or a combination of the above. Obstacle inequality constraints, control inequality constraints can be transformed into equality constraints.

6) able to solve any terminal conditions, any performance indices [3, 13, 14, 15, 18, 21]: In various applications, various performance indices need to be implemented. For example, time, distance, energy or a combination of the above. Point to point task has different initial and final states in applications.

7) has the potential to fully utilize computer power as the computer industries grow in the near future: Several years from now, computers can be many times faster in CPU. We don't reject any ideas which consume more CPU time than we can afford today. On the contrary, we encourage numerical method that fully utilizes the modern scientific computing concepts. Provided we have infinitesimal small stepsize and infinite digits, and we have sufficient CPU time on computers, this calculus of variation approach generates solutions which satisfy exact necessary conditions. Sufficient conditions can also be checked.

1.6 Drawbacks

There are certainly some drawbacks: 1) Minimum distance finding problem within the optimal control problem at each time stage is difficult. For more complex manipulators and general obstacles, the minimum distance can be found through optical devices or solid modeling techniques. In common sense, a human being has to sense (by eyes) obstacles and potential of collision before he/she can think about avoidance. 2) Due to the consumption of huge amount of CPU time, the method is good for off-line programming but is not yet ready for real-time, on-line applications. This is a tradeoff for a new promising technique.

1.7 Contents

A brief description of the dynamic systems and constraints are given in section 2. Section 3 contains the obstacles avoidance schemes. Section 4 contains the optimal control theory. Section 5 contains the sequential gradient restoration algorithm. In Section 6, numerical examples of a two degree-of-freedom robotic manipulator are demonstrated. The insight of collision-free minimum-time motion are shown in captions and tables. Finally, discussion is in section 7, conclusion and prospective research are presented in section 8. Appendix A illustrates the kinematics of a two-link manipulator example.

2. Dynamic Systems and Constraints

The highly nonlinear dynamic equations and inequality control constraints and/or inequality state constraints are also the main difficulties of optimal control on robotics.

2.1 Dynamics

Under the assumption that the links are uniform rods of mass m_i at the mass center, of moment of inertia I_i, of length l_i, respectively, i is the number of the link. The gravity g is acting parallel to the negative y-axis direction. The dynamical equations can be derived by means of Newton-Euler (Lagrange-Euler) equations [37, 38, 39] or symbolic method [40] and expressed in general as:

$$\tau = M(\theta)\alpha + C(\theta,\omega) + G(\theta) \qquad (1)$$

where τ is the vector of applied torques/forces, $M(\theta)$ is the inertial matrix terms of the manipulator, $C(\theta,\omega)$ is the vector of centrifugal and Coriolis terms, $G(\theta)$ is the vector of gravity terms. For example, a two-link manipulator in Fig. 1. [16]:

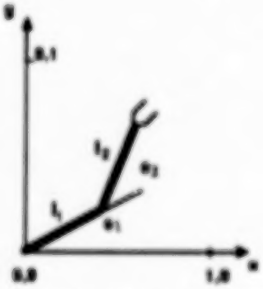


Fig.1. Two-link robotic manipulator
Link 1 of m_1, l_1, I_1
Link 2 of m_2, l_2, I_2

$$M(\theta)=\begin{pmatrix} I_1+I_2+(m_1)_1^2+m_2(l_2)^2)/4+m_2l_1^2+m_2l_1(l_2\cos\theta_2 & I_2+(m_2(l_2)^2)/4+(m_2l_1(l_2\cos\theta_2)/2 \\ I_2+(m_2(l_2)^2)/4+(m_2l_1(l_2\cos\theta_2)/2 & I_2+(m_2(l_2)^2)/4 \qquad I_2 \end{pmatrix} \qquad (2)$$

$$C(\theta,\omega)=\begin{pmatrix} -m_2l_1l_2\sin\theta_2(\omega_2)^2/2-m_2l_1l_2\sin\theta_2(\omega_1)(\omega_2) \\ m_2l_1l_2\sin\theta_2(\omega_1)^2/2 \end{pmatrix} \qquad (3)$$

$$G(\theta)=g\begin{pmatrix} m_2l_2\cos(\theta_1+\theta_2)/2+l_1(m_1/2+m_2)\cos\theta_1 \\ m_2l_2\cos(\theta_1+\theta_2)/2 \end{pmatrix} \qquad (4)$$

One can see that these highly nonlinear terms are functions of the joint velocities and angles. $\theta_i, \omega_i, \alpha_i$ are relative angle, angular velocity, and angular acceleration of link i respectively.

2.2 Control Systems and Inequality Control Constraints

Robotics dynamic system can be formulated in two ways:

2.2.1 Kinematical Formulation

Kinematical formulation is practical in most cases, specially when the model reference dynamic parameters are not known in advance. For example, the payload is never known ahead; or for safety reason that the inertia force caused by acceleration of the robots shall be limited. In kinematical formulation, the control system is as follows:

$$\dot{\theta} = \omega \qquad (5)$$

$$\dot{\omega} = \alpha \qquad (6)$$

θ, ω, α are vector of state variables. Once the states in joint space of the manipulator are known, we can compute the joint torques which are required to balance the reaction forces/moments acting on the links. The physical inequality constraints imposed on the robot in this study are joint acceleration bounds [20]. With these constraints, we can limit the torques which are related to the joint space configuration. In terms,

$$|\alpha| \le C, \qquad (7)$$

Via the following variable transformation, the joint acceleration can be limited within the bounds

$$\alpha = C\sin(u), \qquad (8)$$

C is vector of upper bounds of the absolute acceleration in joint space. u is vector of the new control variable.

2.2.2 Dynamical Formulation

If we know the model reference system in advance, in dynamical formulation, the control system is as follows:

$$\dot{\theta} = \omega \qquad (9)$$

$$\dot{\omega} = M^{-1}(\tau - C(\theta,\omega) - G(\theta)) \qquad (10)$$

θ, ω, α are state variables. In this formulation, we assume the dynamic parameters in matrices M, C, G, are known. The matrix M is always both "symmetric and positive definite" [38], therefore always invertible. The physical inequality constraints imposed on the robot in this formulation are joint torque/force bounds. With these constraints, we can limit the torques in the actuator space configuration. In terms,

$$|\tau| \le C, \qquad (11)$$

Via the following variable transformation, the joint torque can be limited within the bounds

$$\tau = C\sin(u), \qquad (12)$$

C is vector of upper bounds of the absolute torque in actuator space. u is vector of the new control variable.

2.3 Equality Constraints

In some cases, the end-effector has to follow a specified path, or the orientation of the arm in the motion is specified and fixed, for example, the robot arm is holding a flash light moving along a specified path, then, the degree of freedom is reduced by the number of constraints. One or more state constraints have to be added in Cartesian space, then converted into joint space. The system is solved with replacement of the algebraic equation into the state variables according to the constraints.

3. Obstacles Avoidance Schemes

By definition, obstacles can be avoidable or unavoidable for a fixed configuration. Configuration has to be fixed in one task to avoid excess excursion and changing kinematics. For examples, in Fig. 2., the obstacle is away from the robot chassis but within the work envelope. That is considered as avoidable. In Fig. 3., obstacles are too close to the robot and there is no space for feasibly moving the robot arm through the obstacle environment. This is considered as unavoidable.

For simplicity, each obstacle is put into an ellipsoid. It is a little wasteful to put an obstacle which is not necessarily in ellipsoid shape into an ellipse. The advantage is the ellipsoid parameters can be changed to

shapes the end into the figure of the obstacle without wasting too much space. Collision avoidance can be achieved by continuously controlling the closest point on the arm to the obstacles.



Fig. 2. Avoidable Obstacle



Fig. 3. Unavoidable Obstacles

3.1 Stains Inequality Constraints

Let Q_i denote the i-th obstacle ellipsoid function among m obstacles. The obstacle constraint is:

$$Q_i = a_i(x-x_2)^2 + b_i(x-x_2)(y-y_2) + c_i(y-y_2)^2 + f_i \geq 0 \qquad (13)$$

For collision avoidance, it is a must that at all times, for the closest point on the arm,

$$Q_i \geq 0 \qquad (14)$$

3.2 Distance Functions

Distance function D_i is defined as the function $Q_i(x,y)$ from the closest point (x, y) on the arm to the i-th obstacle.

The position on each link can be identified by

$$x=x_2+\lambda(x_2-x_1); \quad y=y_2+\lambda(y_2-y_1) \qquad (15)$$

in terms. x, y are functions of a parameter λ. x_1, y_1, x_2, y_2 are Cartesian coordinates at end points of the links.

Substituting (x, y), $Q_i(x, y)$ becomes a function of parameter λ. To find the closest point from the arm to the obstacle, we take differentiation and find minimum Q_i versus λ

$$\frac{dQ_i}{d\lambda} = 0, \qquad (16)$$

thus, D_i is chosen among the closest points on links to the obstacle. In most of the avoidable obstacle cases, the closest point happens to be on the forearm at end-effector.

When $D_i=0$, it means the arm touches the i-th obstacle at the closest point. When D_i is infinitesimally small, it means the arm grazes the i-th obstacle.

3.3 Virtual Potential Field Penalty Function Method (PF)

The penalty function P_i is defined as

$$g_i = \exp(D_i/\varepsilon_i)-1 \qquad (17)$$

$$P_i = \frac{1}{g_i} \qquad g_i \geq \varepsilon \qquad (18)$$

$$P_i = \Gamma \qquad g_i < \varepsilon \qquad (19)$$

ε_i is a small number which denotes the dramatic steepness factor between the trajectory and the i-th obstacle where the penalty becomes active. Γ is a huge number on the edge of the precision boundary that causes computer overflow. ε is a tiny number on the edge of the precision boundary that causes computer underflow. The merit of this infinite penalty function is by choosing a small number ε_i, one can define how close the trajectory is allowed to clear the i-th obstacle. By increasing the value ε_i, one can supplant the steepness of the penalty function so the trajectory will never get into the obstacles' forbidden area. As D increases, P sharply decreases, i.e. almost no penalty in farther distance; as D decreases, P dramatically increases, i.e. a sudden increase of a penalty barrier in the goal function for obstacle avoidance. As soon as P dominates the goal function, the problem changes from a minimum-goal one into an obstacle avoidance one. See Fig. 4. as following:

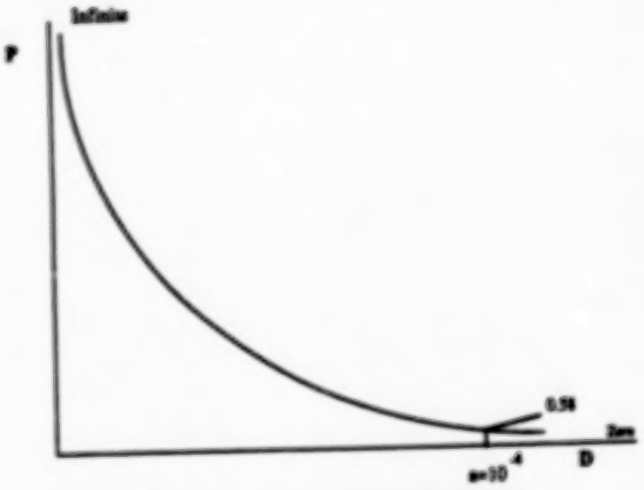


Fig. 4. Penalty function versus distance function diagram

For this optimal control system, we have formulated θ, ω, α as state variables x; α as control variables. For the time-optimal problem, the performance index is:

$$I = \alpha + \int_0^1 \alpha \, P_d(\tau) \, d\tau \qquad (20)$$

$$P_d(t) = \sum_{i=1}^{m} P_i(\tau) \qquad (21)$$

Boundary conditions are initial states x(0), and final states x(1) of specified values. Once the states and the controls are computed, the required reacting torques T can be solved from Eq. (1).

3.4 Violation Compensation Penalty Function Method (PI)

The penalty function P_i is defined as

$$P_i = 0 \qquad D_i \geq 0 \qquad (22)$$

$$P_i = -(D_i - \varepsilon)^2 \qquad D_i < 0 \qquad (23)$$

ε is a small number. This penalty is a negative compensation function via the distance function. The merit of this penalty function is to force the violation of the obstacle constrain out as the negative sign indicates.

For this optimal control system, we have the same state variables and control variables as above. For the time-optimal problem, the new performance index is:

$$I = \alpha + \int_0^1 \alpha \, P_d(\tau) \, d\tau \qquad (24)$$

$$P_d(t) = \sum_{i=1}^{m} W_i \, P_i(\tau) \qquad (25)$$

W_i is a weighing factor for the corresponding penalty function.

3.5 Variables Transformation Method (PI)

For collision avoidance, by introducing a new variable s.

$$Q_i = a_i(x-x_2)^2 + b_i(x-x_2)(y-y_2) + c_i(y-y_2)^2 + f_i = s^2 \qquad (26)$$

$$\dot{s} = [a_i(x-x_2)\dot{x} + b_i(x-x_2)\frac{\dot{x}}{2}(y-y_2) + b_i(y-y_2)\frac{\dot{x}}{2}(x-x_2) + c_i(y-y_2)\dot{y}]/s \qquad (27)$$

where $\dot{x}$, $\dot{y}$, are the time differentials of x, y. We add one or more differential constraints to the control system. For this optimal control system, we have formulated $\theta, \omega, \alpha, s$ as state variables x; α as control variables. For the time-optimal problem, the performance index is: $I = \alpha$

3.6 Time Scaling

In the above systems, time has been normalized from $t_{initial}=0$ to $t_{final}=1$ via the following transformation: $t=\sigma\tau$, i.e., $dt=d\tau\cdot\sigma$ is a parameter which represents the final time.

3.7 Minimum Distance Problem

For the minimum-distance problem of the end-effector in Cartesian space, the performance index can be replaced by:

$$I=\int_0^1 \sigma\,(v_x^2+v_y^2)^{1/2}\,d\tau=\int_0^1 \sigma\,P_d(\tau)\,d\tau \qquad (28)$$

3.8 Primal Formulation

Optimal control has the characteristic of duality [41, 42]. In this study, the sequential gradient restoration algorithm is employed in conjunction with primal formulation.

4. Optimal Control Theory

The optimal control problem [43] is described in general as follows:

With respect to the vectorial state variable $x(t)$, vectorial control variable $u(t)$ and the vectorial parameter π, the problem of minimizing a functional

$$I=\int_0^1 f(t,x,u,\pi)dt+[h(t,x)]_0+[g(t,x)]_1 \qquad (29)$$

subject to differential constraints:

$$\dot{x}-\phi(t,x,u,\pi)=0, \qquad 0\le t\le 1, \qquad (30)$$

initial conditions:

$$[\psi(t,x)]_0=0, \qquad (31)$$

and final conditions:

$$[\psi(t,x)]_1=0. \qquad (32)$$

where f, h, g, are scalar functions, and ϕ, ψ, η are vectorial functions of specified dimensions. t is a independent variable. The subscript 0 denotes the initial point, and the subscript 1 denotes the final point.

Optimality Criteria

By introducing the Lagrange multipliers, the problem shown above can be recast as minimizing the augmented functional J

$$J=I+L \qquad (33)$$

subject to Eqs. (30-32), where L is the Lagrangian functional

$$L=\int_0^1 \lambda^T(\dot{x}-\phi(t,x,u,\pi))dt+(\nu^T\psi)_0+(\mu^T\psi)_1 \qquad (34)$$

The symbols $\lambda(t)$, ν, μ denote Lagrange multipliers of appropriate dimensions associated with the constraints. The superscript symbol ^{T} denotes the transpose of the matrix.

The first-order optimality criteria originated from Pontryagin Extremum Principle for Eqs. (29-34) can be derived from Euler equations in calculus of variation as:

$$\dot{\lambda}-f_x+\phi_x\lambda=0, \qquad 0\le t\le 1, \qquad (35)$$

$$f_u-\phi_u\lambda=0, \qquad 0\le t\le 1, \qquad (36)$$

$$\int_0^1 (f_\pi-\phi_\pi\lambda)dt+(h_\pi+\nu\psi_\pi)_0+(g_\pi+\mu\psi_\pi)_1=0, \qquad (37)$$

$$(\lambda+h_x+\nu\psi_x)_0=0, \qquad (38)$$

$$(\lambda+g_x+\mu\psi_x)_1=0. \qquad (39)$$

In terms, we seek the functions $x(t)$, $u(t)$, π and the multipliers $\lambda(t)$, ν, μ such that the feasibility Eqs. (30-32) and the optimality criteria Eqs. (35-39) are satisfied to certain numerical accuracy.

5. Algorithm

The sequential gradient restoration algorithm, in either the primal formulation or the dual formulation, is an iterative technique which is constructed by a sequence of two-phase suboptimal cycles. Each cycle includes a gradient phase and a restoration phase. In the gradient phase, the value of the augmented functional is decreased in one step, while avoiding excessive constraint violation. In the restoration phase, the value of the constraint error is decreased in one or multiple steps, while avoiding excessive change in the value of the functional. In a complete gradient-restoration cycle, the value of the functional is decreased, while the constraints are satisfied to a pre-selected degree of accuracy. Therefore, a sequence of suboptimal solutions is generated. Each new suboptimal solution is an improvement of the previous one from the point of view for the value of the functional to be minimized. The optimal solution is reached when the optimality error and the constraint error are both satisfied to a certain accuracy. Schematic diagram is shown in Fig. 5:

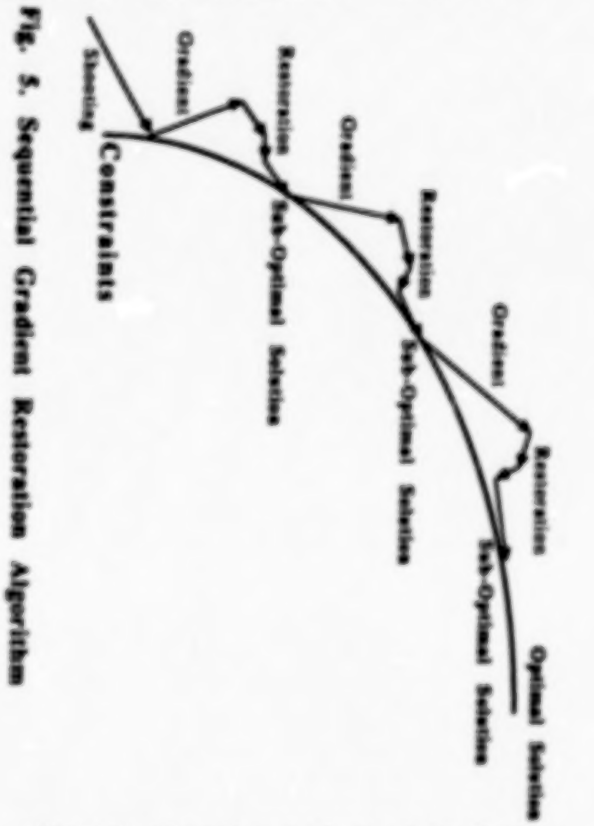


Fig. 5. Sequential Gradient Restoration Algorithm

Let $x(t)$, $u(t)$, π denote the nominal functions; let $\bar{x}(t)$, $\bar{u}(t)$, $\bar{\pi}$, denote the varied functions; let $\Delta x(t)$, $\Delta u(t)$, $\Delta\pi$ denote the displacements leading from the nominal functions to the varied functions. Under the assumption that the displacements $\Delta x(t)$, $\Delta u(t)$, $\Delta\pi$ are linear with respect to α, where $\alpha>0$, and $A(t)$, $B(t)$, C denote the displacements per unit respect, then the following relations can be used for iterations:

$$\bar{x}(t)=x(t)+\Delta x(t)=x(t)+\alpha A(t) \qquad (40)$$

$$\bar{u}(t)=u(t)+\Delta u(t)=u(t)+\alpha B(t) \qquad (41)$$

$$\bar{\pi}=\pi+\Delta\pi=\pi+\alpha C \qquad (42)$$

Thus, each iteration of the gradient phase and the restoration phase involves two distinct operations: (i) the determination of the direction functions $A(t)$, $B(t)$, C, and (ii) the determination of the respect of variation α.

From (40-42) and constraint conditions (30-32), one can derive the following relations from first order variation:

$$\dot{\lambda} - a_x^T A - a_y^T B - a_\theta^T C + \Omega_2(\dot{\delta} - \omega) = 0, \qquad 0 \le t \le 1, \tag{43}$$

$$(a_x^T A + a_y^T C + \Omega_2 a)_0 = 0, \tag{44}$$

$$(a_x^T A + a_y^T C + \Omega_2 a)_1 = 0, \tag{45}$$

and from Eqs. (40-42) and first-order optimality criteria (35-39), one can derive the following relations from first order variation:

$$\dot{\delta} - \Omega_1 \delta_\omega + a_\omega \lambda = 0, \qquad 0 \le t \le 1, \tag{46}$$

$$\dot{\omega} + \Omega_1 \delta_\omega - a_\omega \lambda = 0, \qquad 0 \le t \le 1, \tag{47}$$

$$C + \int_0^1 [-a_\omega \lambda \delta\theta + (w_\omega a)_\theta + (w_\omega a)_\omega + \Omega_1 \delta] \int_0^\theta \zeta_\omega d\theta + (b_\omega)_\theta + (b_\omega)_\omega \} = 0, \tag{48}$$

$$(A - \lambda + w_\omega a + \Omega_1 b_\omega)_0 = 0, \tag{49}$$

$$(\lambda + w_\omega a + \Omega_1 b_\omega)_1 = 0, \tag{50}$$

where, in the gradient phase, $\Omega_1 = 1, \quad \Omega_2 = 0,$ (51)

in the restoration phase, $\Omega_1 = 0, \quad \Omega_2 = 1.$ (52)

The above linear two-point boundary-value problem (LTP-BVP) can be solved for the direction functions A(t), B(t), C, by the method of particular solutions (44, 45).

Stepsize

Eqs. (40-42) define one-parameter functions of the stepsize α. For this parameter, the functionals I, J, P become functions of α as following:

$$I = I(\alpha) \qquad J = J(\alpha) \qquad P = P(\alpha) \tag{53}$$

Then, bisection technique is used for the one-dimension search to find the stepsize, starting from reference stepsize α_g in gradient phase, until (i)

$$J(\alpha) < J(0), \qquad P(\alpha) < P_o, \tag{54}$$

P_o is a preselected number, not necessarily small; and starting from reference stepsize α_r in restoration phase, until (ii)

$$P(\alpha) < P(0) \tag{55}$$

In a complete, successful gradient-restoration cycle, the following condition must be satisfied or the cycle is restarted with reduced stepsize.

$$I_i < I_{i-1} \tag{56}$$

where I_i denotes the value of the functional (29) after current cycle. I_{i-1} denotes the value of the functional (29) after the previous cycle.

Updating suboptimal solution schemes

Once the direction function A(t), B(t), C, and stepsize α are solved, the states, the controls, and the parameters are updated according to Eqs. (40-42).

Summary of Algorithm

Let P be the square sum of the error associated with the feasibility Eqs. (1-32), and Q be the square sum of the error associated with the optimality criteria Eqs. (35-39), then

$$P = \int_0^1 N(\dot{\delta} - \omega)^2 d\theta + N(\omega)_0 + N(\omega)_1 \tag{57}$$

$$Q = \int_0^1 N(\dot{\lambda} - f_\delta + a_\omega \lambda)^2 d\theta + \int_0^1 N(I_\omega - a_\omega \lambda) d\theta + N \int_0^1 (I_\omega - a_\omega \lambda)^2 d\theta + N(\lambda + b_\omega + w_\omega a)_0^2 + N(\lambda + b_\omega + w_\omega a)_1^2, \tag{58}$$

where, N denotes quadratic norm operator.

Thus, numerical convergence for optimal solution can be declared when

$$P < \varepsilon_1, \tag{59}$$

$$Q < \varepsilon_2. \tag{60}$$

$\varepsilon_1, \varepsilon_2$ are preselected, small, positive numbers.

The algorithm is started from providing nominal functions of u(t), and π. The nominal functions can be provided arbitrarily, but good nominals help convergence. The nominal controls are provided with a standard shooting method of Modified Quasilinearization Algorithm, followed by solving the nominal states based on nominal controls, to some accuracy of terminal conditions.

Then, the restoration phase is started. Eqs. (43-45) are solved with (52) and search of stepsize in restoration phase. The one or more iteration restoration phase is completed only until Eq. (59) is satisfied. Then, the gradient phase is started. Eqs. (46-50) are solved with (51) and search of stepsize in gradient phase until Eq. (60) is satisfied for only one iteration. The restoration phase is started again. Thus, a sequence of suboptimal solutions is generated. Each new solution is an improvement of the previous one from the point of view for the value of the functional to be minimized. The optimal solution is reached when Inequ. (59-60) are both satisfied.

4. Numerical Examples

Numerical examples for time-optimal control with obstacles avoidance schemes of a two-link robotic manipulator are shown in this section. The numerical and analytical solutions of time-optimal control without obstacles can be refered to [33]. The following physical parameters are taken from Asada [46], Sahar and Hollerbach [16] and Zhang and Wang [30]. The obstacle is put on the midway, or even further inward, of the previous no-obstacle optimal trajectory on purpose. The algorithm can be applied to any degree-of-freedom robots with arbitrarily given physical parameters and boundary conditions.

In joint space,

initial position $(\theta_1, \theta_2)_0$ = (0.22, 0.35)	rad.
final position $(\theta_1, \theta_2)_f$ = (0.8208, 1.4289)	rad.
initial velocity $(\omega_1, \omega_2)_0$ = (0.0, 0.0)	rad/sec.
final velocity $(\omega_1, \omega_2)_f$ = (0.0, 0.0)	rad/sec.
acceleration bounds (C_1, C_2) = (0.5, 1.0)	rad/(sec)^2.
gravity constant g = 9.8	m/(sec)^2.

Link 1
mass m_1 = 30 kg, length l_1 = 0.5 m, moment of inertia I_1 = 3.0 kg(m)^2.
Link 2
mass m_2 = 30 kg, length l_2 = 0.5 m, moment of inertia I_2 = 3.0 kg(m)^2.

The ellipse obstacle is represented by the following equation:
$Q = a_0(x-x_0)^2 + b_0(x-x_0)(y-y_0) + c_0(y-y_0)^2 + f_0 = 0$, where, $f_0 = -(r_0)^2$

The following symbols are used in the table:
P1: Virtual Potential Field Penalty Function Method
P2: Violation Compensation Penalty Function Method
P3: Variable Transformation Method
E : Ellipse Obstacle, C : Circle Obstacle, when $b_0 = 0$

Table 1. Comparison of Obstacle Avoidance Schemes

	(P1-C)	(P2-C)	(P3-C)
x_0 (m)	0.5	0.5	0.5
y_0 (m)	0.76	0.76	0.76
r_0 (m)	0.1	0.1	0.1
a_0	1.0	1.0	1.0
b_0	0.0	0.0	0.0
c_0	1.0	1.0	1.0
ε (m)	10^{-6}	∞	∞
ε (m)	∞	5.0×10^{-6}	∞
W	∞	0.5×10^{12}	∞
minimum time (sec)	3.914	5.071	4.597

∞ denotes there is no such value for the criteria.
5.197 sec is the minimum time without obstacle avoidance.

Table 1. contains the insight of comparison between two penalty function methods and the variables transformation method for circle obstacle (r_0=0). Circle obstacle is centered at (0.5, 0.75) of radius 0.1. Virtual potential field penalty function has only one parameter and generates grate-by trajectory. Violation compensation penalty function is difficult to implement as a clear collision avoidance scheme owing to the infinite combination of two parameters. Variables transformation method avoids obstacle successfully but is too constrained when the robot is away from the obstacles to generate true local optima.

Table 2. Comparison of Different Location of Ellipse and Circle Obstacles.

P1	(E1)	(E2)	(C1)	(C2)
x_0 (m)	0.51	0.50	0.51	0.50
y_0 (m)	0.63	0.76	0.62	0.76
r_0 (m)	0.2	0.2	0.1	0.1
a_0	2.0	2.0	1.0	1.0
b_0	2.0	2.0	0.0	0.0
q_0	2.0	2.0	1.0	1.0
ϵ (m)	10^{-4}	10^{-4}	10^{-4}	10^{-4}
minimum time (sec)	4.332	3.330	3.051	2.914

3.137 sec is the minimum time without obstacle avoidance.

In Table 2., the circle and ellipse obstacle avoidance are listed for side-by-side comparison. As we see, Ellipse (r_0=0.2) is larger in size than Circle (r_0=0.1). Long axis is 45 degree clockwise oriented. Both E1, C1 have the same center location at (0.51, 0.62) and is more close to the robot than E2, C2 which have the same center location at (0.5, 0.76). Owing to the more journey the obstacle avoidance seems, the bigger the obstacle or the closer the obstacle, the longer the minimum-time is, in trajectory.

For the sake of brief, the results corresponds to the first ellipse (E1) in Table 2. are shown in Figs. 6-11.

Fig. 6 contains the optimal trajectory in minimum time
Fig. 7 contains the joint angle profile in minimum time.
Fig. 8 contains the joint velocity profile in minimum time.
Fig. 9 contains the joint acceleration profile in minimum time.
Fig. 10 contains the torque profile in minimum time.
Fig. 11 contains the distance function profile in minimum time.

Table 3. Comparison of Different Radius of Circle Obstacles.

P1	(C1)	(C3)	(C3)
x_0 (m)	1.0	1.0	1.0
y_0 (m)	1.0	1.0	1.0
r_0 (m)	0.4	0.3	0.4
a_0	1.0	1.0	1.0
b_0	0.0	0.0	0.0
q_0	1.0	1.0	1.0
ϵ (m)	10^{-4}	$0.5x10^{-4}$	10^{-4}
minimum time (sec)	3.029	2.696	2.157

(C3) obstacle is outside the previous no-obstacle optimal trajectory.
3.137 sec is the minimum time without obstacle avoidance.

In Table 3., the circle is moved to center location (1, 1). Radius of the circle is varied as (C1, C3, C3)=(0.4, 0.3, 0.4). In C3 case, the obstacle is outside the work envelop of the robot, so the minimum-time for C3 is the same as the one without obstacle. The minimum-time is longer as the obstacle is bigger in C1, C3 cases.

In C3 case, owing to the influence of the existing penalty function, there are two curves overlapped on the trajectory. One curve is for previous optimal trajectory without obstacle avoidance scheme, another one is optimal trajectory with obstacles avoidance scheme and obstacle is outside the work envelop.

6.1 Severe Obstacle Avoidance

In joint space,

initial position $(\theta_1, \theta_2)_I$ = (0.349, 0.628) rad.
final position $(\theta_1, \theta_2)_F$ = (0.497, 0.332) rad.
initial velocity $(\omega_1, \omega_2)_I$ = (0.0, 0.0) rad/sec.
final velocity $(\omega_1, \omega_2)_F$ = (0.0, 0.0) rad/sec.

The following case study shows the arm started near the edge of one side of the obstacle and ended near the edge of another side of the obstacle.

Table 4. Severe Obstacle Avoidance.

	(P1-C)	(P2-C)
x_0 (m)	0.667	0.667
y_0 (m)	0.667	0.667
r_0 (m)	0.1	0.1
a_0	1.0	1.0
b_0	0.0	0.0
q_0	1.0	1.0
ϵ (m)	10^{-3}	—
ϵ (m)	—	$0.3x10^{-1}$
W	—	$0.5x10^{3}$
minimum time (sec)	2.800	2.911

*** denotes there is no such value for the scheme.
1.000 sec is the minimum time without obstacle avoidance.

As we see, minimum-time control is not necessarily related to minimum-distance of the end-effector. The collision avoidance scheme has excellence to move around and avoid severe obstacle.

7. Discussion

The insights of the merit of the optimal obstacles avoidance are shown above in Fig. 6-15. All the obstacle avoidance trajectories have the following characteristics: 1) grazing by the obstacle. 2) trying to achieve previous no-obstacle trajectory at near bang-bang control for minimum-time. (At least one joint bang-bang control is the solution for minimum-time without obstacles avoidance) 3) achieving previous no-obstacle optimal trajectory with collision avoidance scheme in the cases of no-obstacles. 4) being able to move around and avoid the severe obstacle.

Virtual potential penalty function method does not cause obstacles constraints violation or over-constrained situation, is the one and only best method. Violation compensation method is difficult to implement owing to the two weighting factors which causes a little obstacle constraint violation from time to time. Variables transformation method is over-constrainted when the obstacles are away from potential collision. This discussion matches the one in [4] even though the implementation of penalty function is different. This approach also illustrates the experimental results for optimization with inequality and/or equality constraints.

As we can see from the comparison tables, the minimum-time of collision-free optimal trajectory is relatively depended on the size and location of the obstacles. The jerk control can be overcome by achieving a near optimal motion in which the trajectory is farther away from the obstacle and the minimum-time is longer.

More intensive research need to be done on minimum distance finding through optical devices or solid modeling. Since the robot manipulators are usually constructed by connected links, in most of the cases, we can say obstacles avoidance for fixed configuration is equivalent to the end-effector obstacle avoidance even though this statement is not true in general. We have to solve end-effector obstacles avoidance before we solve other type of problems because the object is usually on the grip.

The numerical experiments have been done on IBM AS9000 mainfram and VAX 8800, are also attempted to be done on Macintosh. The CPU time for current research varies from 20 minutes (IBM) to one hour (VAX). The accuracy also varies from machine to machine without very much difference. As the computer industries are growing, the CPU time or accuracy is not a problem for future scientific computation.

The numerical results are constructed by 100 cycles and 300 iterations whichever reached first. The convergence is fast at early stages, it slows down after the sub-optimal solutions come close to the optimal solution. To save computation, one can set up lower limits for cycle, iteration and CPU time, so near-optimal solutions will be generated based on Pontryagin Extremum Principle.

8. Conclusion

In this paper, collision-free optimal motion and trajectory planning for robotic manipulators are solved by a method of sequential gradient restoration algorithm. Numerical examples of a two degree-of-freedom robotic manipulator are demonstrated. The obstacle is put on the midway, or even further inward, of the previous no-obstacle optimal trajectory on purpose. For trying to achieve previous no-obstacle trajectory, the trajectory tangentially grazes by the obstacle and the minimum-time motion successfully avoids the obstacle. The minimum-time is longer for the obstacle avoidance cases than the one without obstacle. All the numerical experiments indicate the obstacles avoidance scheme has the same characteristics which allows the trajectory gets as close to the optimal as possible but barely graze by the obstacle. The weighting and effective point of the penalty can be defined by one parameter which justify the closeness between the trajectory and the obstacle. The trajectory will try to achieve optimization under the obstacles barrier. This is the most outstanding characteristic than other schemes to achieve collision avoidance and also find the optimal motion without extra excursion.

The obstacles avoidance schemes can deal with multiple obstacles in ellipsoid forms by continuously controlling the closest point from the arm to the obstacle using virtual potential fields as penalty functions via distance functions. The algorithm is very promising in solving collision-free optimal control problems for any degree-of-freedom robotic manipulators with any performance indices and mobile robots as well. The minimum-time motion is at least one joint bang-bang control or near bang-bang control with obstacles avoidance, no matter the controls are imposed on angular accelerations or on actuator torques. The minimum-distance trajectory without obstacles is a straight line.

Since this algorithm generates true local minimum solution based on Pontryagin extremum principle, rather than based on approximations, the results provide a benchmark against which any other optimization can be measured.

The perspective research is to investigate the result of optimal solutions for robotic manipulators when the controls are imposed on actuator constraints, and/or with moving obstacles avoidance under different performance indices; and model-reference adaptive optimal feedback control.

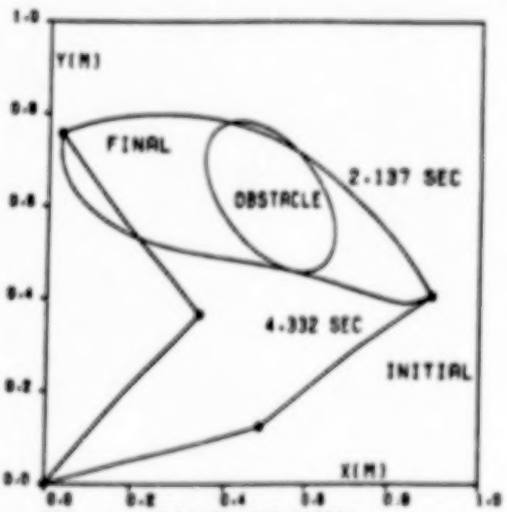


FIG. 6. OPTIMAL TRAJECTORY. MINIMUM TIME= 4.332 SEC .

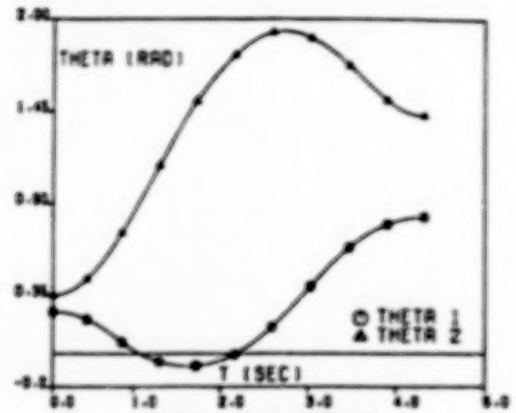


FIG. 7. JOINT ANGLE PROFILE. MINIMUM TIME= 4.332 SEC .

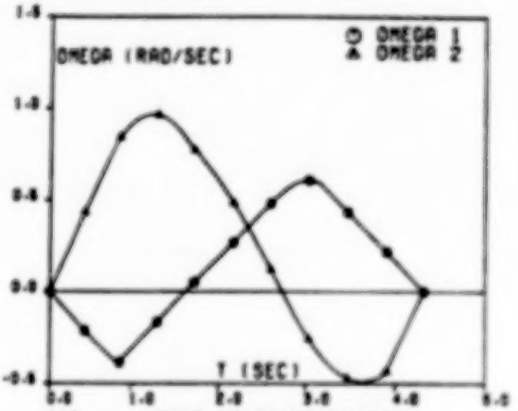


FIG. 8. JOINT VELOCITY PROFILE MINIMUM TIME= 4.332 SEC .

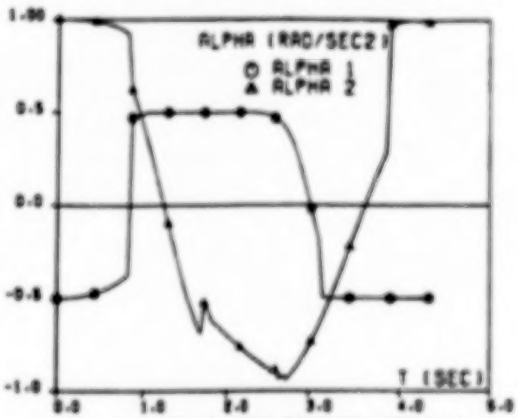


FIG. 9. JOINT ACCELERATION PROFILE. MINIMUM TIME= 4.332 SEC .

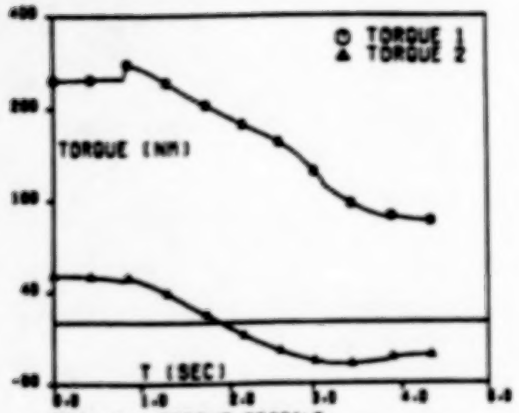


FIG.10. TORQUE PROFILE. MINIMUM TIME: 4.332 SEC .

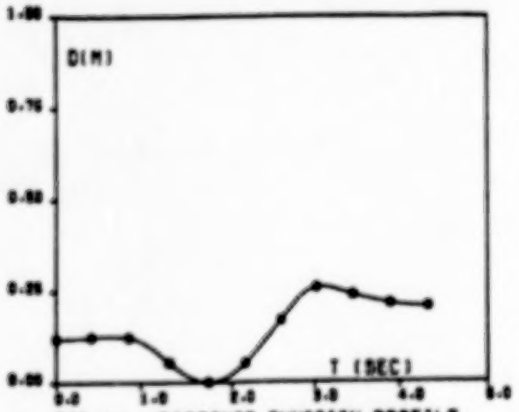


FIG.11. DISTANCE FUNCTION PROFILE. MINIMUM TIME: 4.332 SEC .

Appendix A: Kinematics of a Two-Link Robotic Manipulator

In general, the kinematics, dynamics, control and constraints study of robot can be found in [37]. The kinematical equations are developed by geometrical relationship between Cartesian space and joint space [38]. They can be expressed in general as:

Forward kinematics

$$x(t)=F_1(\theta(t)) \tag{61}$$

$$v(t)=F_2(\theta(t), \omega(t)) \tag{62}$$

$$a(t)=F_3(\theta(t), \omega(t), \alpha(t)) \tag{63}$$

where $x(t)$, $v(t)$, and $a(t)$ are vectors of positions, velocities and accelerations of the end-effector in Cartesian space. $\theta(t)$, $\omega(t)$, and $\alpha(t)$ are vectors of angles, angular velocities and angular accelerations in joint space.

F are functions. For a two-link planar robotic manipulator [40] (Fig.1.):

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} l_1\cos\theta_1+l_2\cos(\theta_1+\theta_2) \\ l_1\sin\theta_1+l_2\sin(\theta_1+\theta_2) \end{pmatrix} \tag{64}$$

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} -l_1\sin\theta_1-l_2\sin(\theta_1+\theta_2) & -l_2\sin(\theta_1+\theta_2) \\ l_1\cos\theta_1+l_2\cos(\theta_1+\theta_2) & l_2\cos(\theta_1+\theta_2) \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} \tag{65}$$

$$\begin{pmatrix} a_x \\ a_y \end{pmatrix} = \begin{pmatrix} -l_1\sin\theta_1 & -l_2\sin(\theta_1+\theta_2) \\ l_1\cos\theta_1 & l_2\cos(\theta_1+\theta_2) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_1+\alpha_2 \end{pmatrix} - \begin{pmatrix} l_1\cos\theta_1 & l_2\cos(\theta_1+\theta_2) \\ l_1\sin\theta_1 & l_2\sin(\theta_1+\theta_2) \end{pmatrix} \begin{pmatrix} \omega_1^2 \\ (\omega_1+\omega_2)^2 \end{pmatrix} \tag{66}$$

Inverse kinematics

$$\theta(t)=G_1(x(t)) \tag{67}$$

$$\omega(t)=G_2(x(t), v(t)) \tag{68}$$

$$\alpha(t)=G_3(x(t), v(t), a(t)) \tag{69}$$

G are functions which depend on the configuration (like: elbow-down). For a two-link robotic manipulator in elbow-down position (Fig.1.):

$$\theta_1=\tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{l_2\sin\theta_2}{l_1+l_2\cos\theta_2}\right), \quad \text{where,} \tag{70}$$

$$\theta_2=\cos^{-1}\left(\frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}\right) \tag{71}$$

$$\begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} = \frac{1}{l_1l_2\sin\theta_2} \begin{pmatrix} l_2\cos(\theta_1+\theta_2) & l_2\sin(\theta_1+\theta_2) \\ -l_1\cos\theta_1-l_2\cos(\theta_1+\theta_2) & -l_1\sin\theta_1-l_2\sin(\theta_1+\theta_2) \end{pmatrix} \times \begin{pmatrix} v_x \\ v_y \end{pmatrix} \tag{72}$$

$$\begin{pmatrix} \alpha_1 \\ \alpha_1+\alpha_2 \end{pmatrix} = \frac{1}{l_1l_2\sin\theta_2} \begin{pmatrix} l_2\cos(\theta_1+\theta_2) & l_2\sin(\theta_1+\theta_2) \\ -l_1\cos\theta_1 & -l_1\sin\theta_1 \end{pmatrix} \begin{pmatrix} a_x \\ a_y \end{pmatrix} + \frac{1}{l_1l_2\sin\theta_2} \begin{pmatrix} l_1l_2\cos\theta_2 & l_2^2 \\ -l_1^2 & -l_1l_2\cos\theta_2 \end{pmatrix} \begin{pmatrix} \omega_1^2 \\ (\omega_1+\omega_2)^2 \end{pmatrix} \tag{73}$$

References

1. Holcomb, L.B., Montemerlo, M.D., "The NASA Automation and Robotics Technology Program", Acta Astronautica, Vol. 16, 1987, pp. 11-18.
2. Lemor, E.J., "Robots on the Space Station", Aerospace America, June 1987, pp. 42-45.
3. Kahn, M.E., Roth, B., "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains", Journal of Dynamic Systems, Measurement, and Control, Vol. 93, 1971, pp. 164-172.
4. Gilbert, E.G., Johnson, D.W. "Distance Functions and Their Application to Robot Path Planning in the Presence of Obstacles" IEEE Journal of Robotics and Automation, Vol. RA-1, No. 1, March, 1985, pp. 21-30.
5. Geering, H.P., Guzzella, L., Hepner, S.A.R., Onder, C.H., "Time-Optimal Motions of Robots in Assembly Tasks", IEEE Transactions on Automatic Control, Vol. AC-31, 1986, pp. 512-518.
6. Kim, Y.T., Jamshidi, M., Shahinpoor, M., "Near-Optimum Control of a Robot Manipulator", International Journal of Robotics and Automation, Vol. 2, 1987, pp. 15-20.
7. Ozaki H., Mohri, A., "Planning of Collision-Free Movements of a Manipulator with Dynamic Constraints", Robotica, Vol. 4, 1986, pp. 163-169.
8. Vukobratovic, M., Kircanski, M., "A Method for Optimal Synthesis of Manipulation Robot Trajectories", Journal of Dynamic Systems, Measurement, and Control, Vol. 104, 1982, pp. 188-193.
9. Singh, S., Leu, M.C., "Optimal Trajectory Generation for Robotic Manipulators Using Dynamic Programming", Journal of Dynamic Systems, Measurement, and Control, Vol. 109, 1987, pp. 88-96.
10. Shin, K.G., McKay, N.D., "A Dynamic Programming Approach to Trajectory Planning of Robotic Manipulators", IEEE Transactions on Automatic Control, Vol. AC-31, 1986, pp. 491-500.
11. Shin, K.G., McKay, N.D., "Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints", IEEE Transactions on Automatic Control, Vol. AC-30, No. 6, 1985, pp. 531-541.
12. Shin, K.G., McKay, N.D., "Selection of Near-Minimum Time Geometric Paths for Robotic Manipulators", IEEE Transactions on Automatic Control, Vol. AC-31, 1986, pp. 501-511.
13. Bobrow, J.E., Dubowsky, S., Gibson, J.S., "On the Optimal Control of Robotic Manipulators with Actuator Constraints", Proceedings of the American Control Conference, San Francisco, California, June 22-24, 1983, pp. 782-787.
14. Dubowsky, S., Norris, M.A., Shiller, Z., "Time Optimal Trajectory Planning for Robotic Manipulators with Obstacle Avoidance: A CAD Approach" IEEE International Conference on Robotics and Automation, San Francisco, CA, April 1986, pp. 1906-1912.
15. Rajan, V.T., "Minimum Time Trajectory Planning", Proceedings of the IEEE International Conference on Robotics and Automation, St. Louis, Missouri, March 25-28, 1985, pp. 759-764.
16. Sahar, G., Hollerbach, J.M., "Planning of Minimum-Time Trajectories for Robot Arms", The International Journal of Robotics Research, Vol. 5, No. 3, 1986, pp. 90-100.
17. Lath, J.Y.S., Lin, C.S., "Optimum Path Planning for Mechanical Manipulators", Journal of Dynamic Systems, Measurement, and Control, Vol. 103, June 1981, pp. 142-151.
18. Weinreb, A., Bryson, A.E., Jr., "Minimum-Time Control of a Two-Link Robot Arm", IFAC Control Applications of Nonlinear Programming and Optimization, Capri, Italy, 1985, pp. 195-199.

19. Meier E.B., Bryson A.E., Jr., "An Efficient Algorithm for Time Optimal Control of a Two-Link Manipulator", AIAA Conference on Guidance and Control, Monterey, CA, August 1987, pp. 204-212.
20. Zhang, W., Wang, P.K.C., "Collision-Free Time-Optimal Control of a Two-Link Manipulator", International Journal of Robotics and Automation, Vol. 1, No. 3, 1986, pp. 96-104.
21. Bobrow, J.E., "Optimal Robot Path Planning Using the Minimum-Time Criterion" IEEE Journal of Robotics and Automation, Vol. 4, No. 4, August 1988, pp. 443-450.
22. Wang, *J.J.*, "Optimal Control of Robotic Manipulators", International Journal of Robotics and Automation, Vol. 4, No. 1, 1989, pp. 27-35.
23. Vetterling W.T., Teukolsky S.A., Press, W.H., Flannery, B.P., Numerical Recipes: The Art of Scientific Computing, Cambridge University Press, 1988.
24. Kelley, H. J., "Gradient Theory of Optimal Flight Paths", ARS Journal, Vol. 30, No. 10, 1960.
25. Gonzalez, S., Miele, A., "Sequential Gradient Restoration Algorithm for Optimal Control Problems with General Boundary Conditions", Journal of Optimization Theory and Applications, Vol. 26, No. 3, 1978, pp. 395-425.
26. Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", The International Journal of Robotics Research, Vol. 5, No. 1, 1986, pp. 90-98.
27. Miele, A., Venkataraman, P., "Optimal Trajectories for Aeroassisted Orbital Transfer", Acta Astronautica, Vol. 11, Nos. 7-8, 1984, pp. 423-433.
28. Miele, A., Basapur, V.K., Mease, K.D., "Nearly-Grazing Optimal Trajectories for Aeroassisted Orbital Transfer", Journal of the Astronautical Sciences, Vol. 34, No. 1, 1986, pp. 3-18.
29. Miele, A., Basapur, V.K., Lee, W.Y., "Optimal Trajectories for Aeroassisted Coplanar Orbital Transfer", Journal of Optimization Theory and Applications, Vol. 52, No. 1, 1987, pp. 1-24.
30. Miele, A., Wang, T., Melvin, W.W., "Optimal Take-Off Trajectories in the Presence of Windshear", Journal of Optimization Theory and Applications, Vol. 49, No. 1, 1986, pp. 1-45.
31. Wang, *J.J.*, "Near-Optimal Guidance for Abort Landing Flight in a Windshear", M.S. Thesis, Rice University, Houston, TX, December 1987.
32. Miele, A., Wang, T., Wang, H., Melvin, W.W., "Optimal Penetration Landing Trajectories in the Presence of Windshear", Journal of Optimization Theory and Applications, Vol. 57, No. 1, 1988, pp. 1-40.
33. Lam., T.T., Bayazitoglu, Y., "Solution to the Orr-Sommerfeld Equation for Liquid Film Flowing Down an Inclined Plane: An Optimal Approach", International Journal for Numerical Methods in Fluids, Vol. 6, 1986, pp. 883-894.
34. Bayazitoglu, Y., Lam., T.T., "Marangoni Convection in Radiating Fluids", Journal of Heat Transfer, Vol. 109, No. 3, 1987.
35. Wang, *J.J.*, "Optimal Control of Robotic Manipulators with Actuator Constraints", AAC Professionals, Science, Engineering and Technology Seminars, Houston, TX, May 28-29, 1988
36. Wang, *J.J.*, "Optimal Control of Robotic Manipulators with Obstacles Avoidance", SIAM Conference on Control in the 90's, San Francisco, CA, May 17-19, 1989
37. Vukobratovic, M., et al, "Introduction to Robotics", Springer-Verlag, New York, 1989.
38. Wolovich, W.A., Robotics: Basic Analysis and Design, CBS College Publishing, New York, NY, 1987.
39. Brady, M., Hollerbach, J.M., Johnson, T.L., Lozano-Perez, T., Mason, M.T., Robot Motion: Planning and Control, The MIT Press, Cambridge, Massachusetts, 1982.
40. Cheng, P.Y., Wang, C.I., Chen, C.K., "Symbolic Derivation of Dynamic Equations of Motion for Robot Manipulators Using Program Symbolic Method", IEEE Journal of Robotics and Automation, Vol. 4, No. 6, December 1988, pp. 599-609.
41. Rockafellar, R.T., "Duality in Optimal Control", Mathematical Control Theory, Edited by W. A. Coppel, Published by Springer-Verlag, New York, NY, 1978, pp. 219-257.
42. Miele, A., Wang, T., "Primal-Dual Properties of Sequential Gradient Restoration Algorithms for Optimal Control Problems", Part 2, General Problems, Journal of Mathematical Analysis and Applications, Vol. 119, Nos. 1-2, 1986, pp. 21-54.
43. Bryson, A.E., Jr., Ho, Y.C., Applied Optimal Control, Blaisdell Publishing Company, Waltham, Massachusetts, 1969.
44. Miele, A., "Method of Particular Solutions for Linear, Two-Point Boundary-Value Problems, Journal of Optimization Theory and Applications, Vol. 2, No. 4, 1968, pp. 260-273.
45. Miele, A., Iyer, R.R., "General Technique for Solving Nonlinear, Two-Point Boundary-Value Problems via the Method of Particular Solutions", Journal of Optimization Theory and Applications, Vol. 5, No. 5, 1970, pp. 382-399.
46. Azuda, H., "Dynamic Analysis and Design of Robot Manipulators Using Inertia Ellipsoids", Proceedings of the IEEE Computer Society 1st International Conference on Robotics, Atlanta, Georgia, March 13-15, 1984, pp. 94-102.

Jyhshing Jack Wang had his B. S. degree from National Cheng Kung University Taiwan, and M. S. from Rice University Houston, both in mechanical engineering. His interested research fields include: robotics, dynamic systems & control, optimization, optimal Control, automation, control systems, flight mechanics, aerospace engineering applications, numerical methods, computer simulation, CAD, CAE. He currently works for The M. W. Kellogg Company in Houston, TX.
TEL: (713) 960-5000 (O), (713) 784-6751 (H), Telex: 166585,
Fax: (713) 960-2052, EOS, Jack Wang

Visualization of Three Dimensional Data

Stephen R. Ellis[1], Stephen Smith,[2] Selim Hacisalihzade[3]
NASA Ames Research Center
Moffett Field, CA 94035

Introduction

A number of investigations and reviews of the characteristics of the virtual space perceived in pictures have been conducted recently (Rosinski et al.,1980; Sedgwick,1986; McGreevy and Ellis,1986; Grunwald and Ellis, 1986; Ellis, Smith and McGreevy,1987; Barfield, Sandford, and Foley, 1989). Despite the fact that the pictures considered were not stereoscopic, viewers typically were reported to develop a clear sense that the pictured objects were laid out in a virtual space. Quantitative characterization of the metrics of the viewer's perceived space will advance our understanding of picture perception and assist the design of displays for aircraft and spacecraft. The objective of the following research is to characterize patterns of errors observers make when relating the judged the exocentric direction of a target presented on a perspective display to their egocentric sense of visual direction. This type of spatial task is commonly faced by operators of telerobotic systems when using a map-like display of their workspace to determine the visual location and orientation of objects seen by direct view. It is also essentially the same task as faced by an aircraft pilot using a cockpit perspective traffic display of his surrounding airspace to locate traffic out his windows.

Previous studies of the error pattern in direction judgements have focused on exocentric judgements for which the subjects indicated their estimates of the target position by adjusting dials to show a target's azimuth and elevation with respect to a reference direction vector (See Figure 1).

This response may be described as exocentric since the dial's frame of reference is external to the observer and contrasts with egocentric judgements in which target position is indicated with respect to a body-referenced coordinate system Accordingly, in order to test the generality of reported biases in estimating azimuth and elevation with exocentric judgements, it is useful to examine the same exocentric task but request the subjects to make egocentric judgements.

For this new response the observer adjusts the visual direction of head-mounted light cursor to indicate his sense of the target's depicted azimuth and elevation with respect to a reference position and reference direction. This response will explicitly test the generality of previously reported bias in which exocentric directions are judged to be away from a reference straight ahead. This bias may be attributed to errors in the subjects ability to determine the view direction used to generate the display (McGreevy and Ellis, 1986; Grunwald and Ellis, 1986; Ellis, Smith, Grunwald, and McGreevy, 1989). Furthermore, use of an egocentric response such as visual direction provides a more natural response than a dial adjustment. In a sense we ask the subjects to imagine themselves oriented in the virtual space along a particular direction vector and then imagine where they would have to look to see the target.

---

[1] NASA Ames Research Center and U.C. Berkeley
[2]Sterling Software, Palo Alto, CA
[3]NRC Research Associate, NASA Ames Research Center

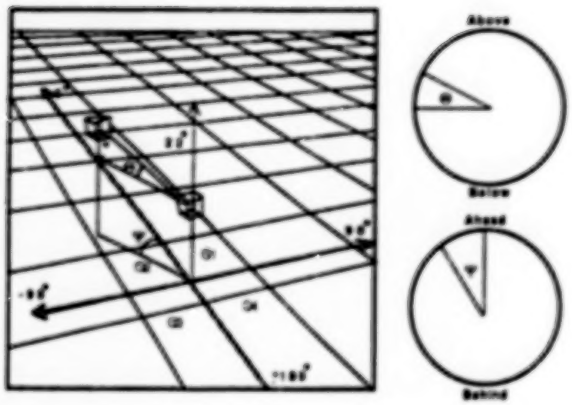


Figure 1. Schematic illustration of the direction judgement task. The subject adjusted the angles ▾ and ● shown on the dials at the right until they appeared equal to the azimuth angle ▾ and the elevation angle ● of the target cube relative to reference at the center. Dotted lines, labels and arrows did not appear on the map display.

Methods

Two groups of 5 subjects participated as independent groups in two experiments. The subjects were male laboratory personnel ranging in age from 20 to 43 who were unfamiliar with the purpose of the experiment.

The experiments were conducted inside a 1.5 m planetarium dome that served as a projection surface for a head mounted, light pointer which projected a red filament image shaped as a 1.5° chevron onto the dome's surface (light from a 3v flashlight bulb through a Wratten #25 filter). The subject's head position was sensed by a Polhemus electromagnetic head tracker attached to an nonmetalic modified welder's helmet approximately 11 cm above the head. The head tracker was independently calibrated against 28 theodolite-positioned, reference markers which were visible during calibration but not during testing.

The subjects were presented with an exocentric judgement task generated by a PDP 11/40 - Evans & Sutherland PS I graphics system. The images used were similar to earlier experiments (McGreevy and Ellis, 1986; Grunwald and Ellis, 1986; Ellis, Smith, Grunwald, and McGreevy, 1989). The major change was the greater yaw of the view direction used to create the images. It was set to a counterclockwise yaw of -35°. Pitch remained -22°. The subjects were seated at the center of the projection in front of the computer calligraphic monitor about 80 cm from the display surface and looked downward into it with a -22 deg. pitch angle matching that of the view vector. The viewport was 17 cm square.

Subjects were first positioned in an adjustable chair so that their head-mounted light cursor pointed to a subjective straight-ahead, eye level that corresponded to the calibration point at 0° pitch and 0° yaw. (See Figure 2) While in this position, a reference reading was taken from the head sensor for all future measurements. The subjects then were instructed to examine a series of automatically, randomly presented displays and to estimate the azimuth and elevation direction of the target with respect to a reference position and direction. Then they were to transfer this judgement to their egocentric frame of reference. They made the judgement by adjusting the pitch and yaw of their head-mounted, light cursor to a position where they would expect to see the target if their head was at the reference position, and initially aligned with the reference direction in the displayed virtual space. For most of the judged directions the subjects could not simultaneously see the display and the cursor position, but had to gaze back and forth between them to accomplish the task, generally using head movements for excentricites greater than 15°. After adjusting the cursor, they held their position and moved a toggle switch that signaled the computer to take the data. The data for a 1 sec. period prior to the switch signal were averaged to give a single measurement. Three replications of each position were taken from each subject in a randomized sequence of 64 measurements that took about 2 hours to complete.

The interpretation of the head-direction data is complicated by the different centers of rotation associated with pitch, yaw and roll of the head. Pure yaws did not displace the center of rotation very much and the measured head yaw to the calibrated positions were within 2° of the calibrated angles within ±60° of the straight ahead, the great-

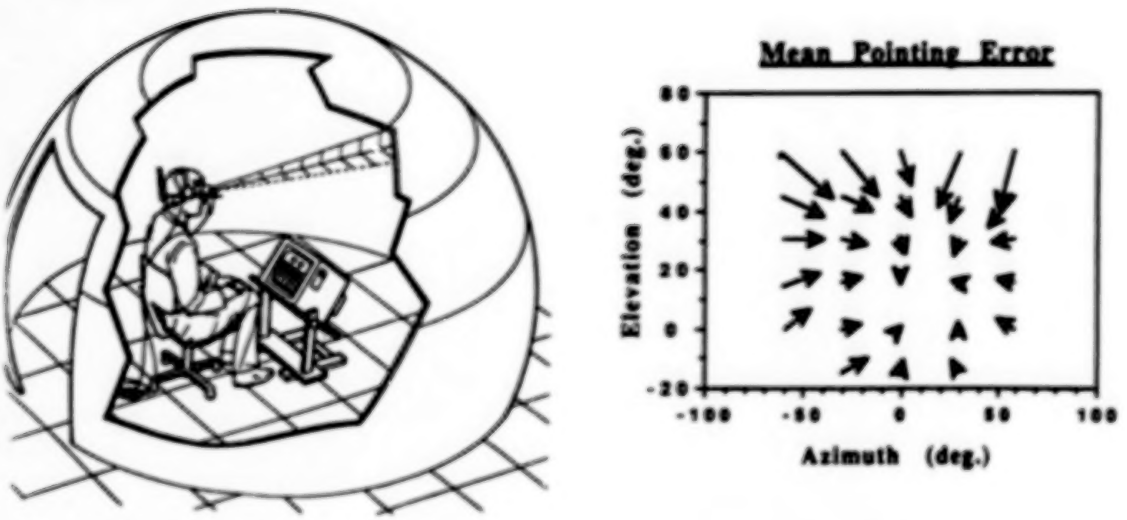


Figure 2. Schematic illustration of the experimental arrangement by which the subject indicated the visual direction at which he would expect to see the target presented on the CRT perspective display if he were positioned at the reference point and aligned with the reference direction. The data in the right portion of the figure represent the average error arcs in a rectangular projection of the forward sphere when both experimental conditions are combined. Each arrow represented the average pitch and yaw error in visual direction to a point at the tail of the arrow.

est deviations being at the most extreme angles. The reason for the residual error was the difficulty of exactly positioning the subject to the calibration reference point. Pitch in contrast tends to be around a moving center of rotation somewhat behind the neck and consequently tends to translate the head upwards and backwards from the initial reference point which was used to provide a straight ahead, level reference for all subsequent measures. Consequently, when the subjects pointed their head-mounted cursors to the extreme pitches, the sensor reading undershot the calibrated value by from 5 to 8 degrees! We have calculated geometrical corrections for the effects of this displacement from the reference point since we could measure it, but generally found that they were small (2-4°) and for reasons discussed below may not in principle be proper to use.

After calibration of the head tracker in the light, the two experiments were conducted in the dark with the CRT display turned down so that only the frame of the monitor was faintly visible to provide an egocentric direction reference. *In one experiment the head cursor was kept on. In the other the cursor was turned off and the subjects had to rely principally on vestibular and proprioceptive cues to "look" to the direction they would expect to see the target.*

Results and Discussion

The results from both experiments were similar and are analyzed together in this summary. Multivariate analysis of variance conducted with BMDP 4V on the elevation and azimuth errors showed that for both judgements the target elevation, target azimuth had statistically reliable effects on both the pitch and the yaw of the errors in the head pointing error. Pitch direction errors: Target Elevation: $F=16.14$ $df=4,5$ $p < .009$; Target Azimuth: $F=7.08$, $df=4,5$ $p<.027$; Yaw direction errors: Target Elevation: ns; Target Azimuth: $F=29.5$ $df=4,32$ $p<.001$. Standard errors for the mean error ranged between 1 and 10 degrees. The main effect of the presence of the light cursor was not significant and did not interact with other independent variables (See Figures 3 and 4)

Since we did not anticipate errors as large as actually measured, we did not use spherical statistics to correct the problems of mapping spherical data into a linear analysis. Since the analysis was conducted on the error data *corrected for wrap-around* of the scale and most of the errors were less than 15 degrees, use of spherical statistics is not likely to substantially change the major results.

The proper method to use to correct for movement of the subject from his calibrated reference position while he positions the cursor depends upon his interpretation of the of the meaning of the cursor position. If he considers its image on the inner surface of the sphere to represent the location of a target cube at that distance, about 1.5m, he would have to introduce parallax corrections to his body-referenced, head direction as he translated with respect to the original reference point so as to keep the cursor on the same place on the sphere as he moved. Alternatively, if he considered, as he in fact was instructed, the cursor image to represent a body-referenced direction toward the target, head displacement in itself would not require adjustment of head direction to keep to cursor properly pointed. This condition is particularly true since he was instructed that the target was at a relatively great distance from the reference cube. For the layouts used, the distance between target and reference was 6m and the viewing distance was modeled at 28m. At this distance the parallax correction for a 5 cm lateral movement would have to be only about 0.5°, comparable to the biological noise associated with head direction. Thus, since the head-angle was measured with respect to a body-referenced straight ahead, correction for head displacement need not be made.

The observed mean body-referenced errors for both experiments are plotted in Figures 3 and 4 as error arcs on a rectangular projection of the response sphere. The pattern shows a tendency to err towards the subject's egocentric straight ahead, but with a significant asymmetry. The results may be interpreted as a composition of errors: 1) the asymmetrical pattern previously reported for exocentric dial responses which is generally away from the straight ahead and 2) a larger but symmetric tendency to overesti-

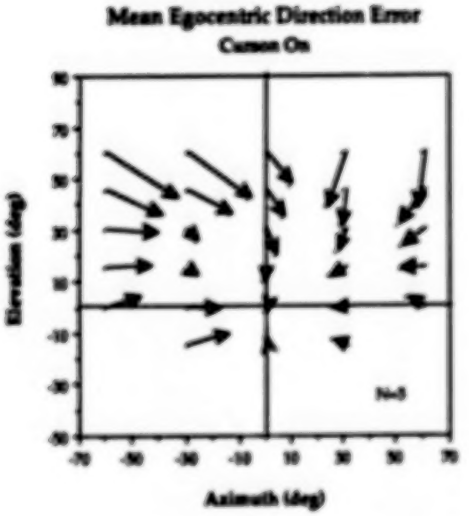


Figure 3. The data in the figure represent the average error arcs in a rectangular projection of the forward sphere for the condition in which the head driven cursor was turned on. Each arrow represents the average pitch and yaw error in visual direction to a point at the tail of the arrow.

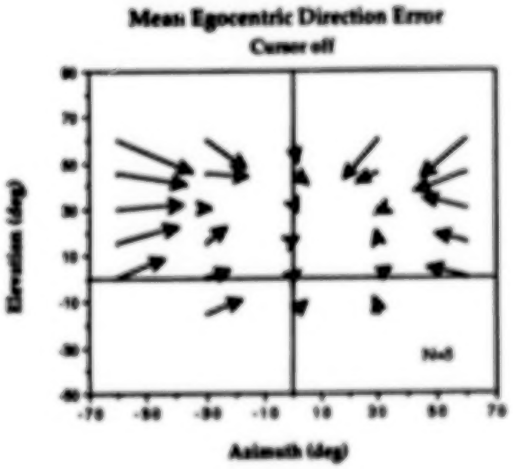


Figure 4. The data in the figure represent the average error arcs in a rectangular projection of the forward sphere for the condition in which the head driven cursor was turned off. Each arrow represents the average pitch and yaw error in visual direction to a point at the tail of the arrow.

mate the extent of the gaze direction indicated by the head mounted cursor. Over estimates like this have been reported by Biguer et al. (1984) for hand pointing to visual target and for head pointing to brief

auditory targets (Perrott, Ambarsoom, and Tucker, 1987). In the case of hand pointing without visual feedback of pointing error such overestimates result in overshoot errors. In the case of head pointing without pointing error feedback, the overestimates result in undershoot errors similar to those observed.

The observation that the errors were not effected by turning off the light cursor supports the idea that one source of error arises from the proprioceptive and vestibular estimate of head rotation. But whether the phenomena is truly one of gaze remains to be determined by future experiments examining gaze angles produced by different combinations of eye and head angles. The results of the current study clearly show however, the visual direction is a significantly biased metric of virtual space presented by flat panel perspective displays. Modeling and explanation of the causes of the observed biases will allow design of compensated perspective displays.

References

Barfield W., Sandford, J. and Foley, J. (1989) The mental rotation and perceived realism of computer generated three dimensional images. *International Journal of Man-machine studies* (in press).

Biguer, B., Prablanc, C., Jeannerod, M. (1984) The contribution of eye and head movements in hand pointing accuracy. *Experimental Brain Research, 55*, 462-469.

Grunwald, Arthur, & Ellis, Stephen R. (1986) Spatial orientation by familiarity cues. *Proceedings of the 6th European Annual Conference on Manual Control*, June, 1986, University of Keele, Great Britain.

Ellis, Stephen R., Smith, Stephen, McGreevy, Michael W. (1987) Distortions of perceived visual directions out of pictures. *Perception and Psychophysics*, 42, 535-544.

Mardia, K.V. (1972) *The statistics of directional data* Academic Press, New York.

McGreevy, Michael W. and Ellis, Stephen R. (1986) The effect of perspective geometry on judged direction. *Human Factors*, 28, 439-456.

Perrott, David (1987) Changes in head position as a measure of auditory localization performance: auditory psychomotor coordination under monaural and binaural listening conditions. *Journal of the Acoustical Society of America, 82*, 1637-1645.

Rosinski , R.R., Mulholland, T, Degelman, D., and Faber, J. (1980) Picture perception: an analysis of compensation. *Perception and Psychophysics, 28*, 512-525.

Sedgwick, H. A. (1986) Space perception in *Handbook of Perception and Human Performance vol 1*. K. R. Boff, L. Kaufman, J.P. Thomas eds, New York: Wiley, 21:19 - 21.

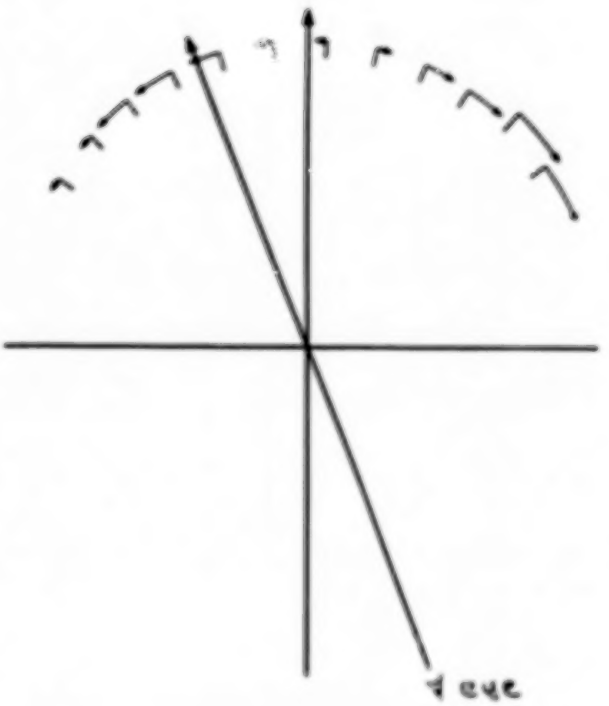


Figure 5. Circular plots for perspective displays in which subjects indicated target azimuth for targets at 0 degrees elevation by adjusting an angle on a dial. The errors are plotted as directed arcs with the tail of each arrow at the correct position of the target. The length of each arrow represents the average error from 8 subjects. Though the viewing azimuth was -22° compared to the -35° used in the current experiments, the conditions are otherwise comparable. The error arcs clearly show a bias away from the straight ahead rather than towards it and also show an asymmetry with greater errors in the right quadrant than in the left. Thus, if this bias were to cancel a larger one, perhaps due to overestimation of gaze direction, that was toward the straight ahead, the resulting bias would be smaller in the right quadrant than in the left. This expected pattern in found the the data for zero degree target orientation in Figures 3 and 4.

REMOTE OPERATION OF AN ORBITAL MANEUVERING VEHICLE IN SIMULATED DOCKING MANEUVERS

Adam R. Brody
Sterling Software
NASA/Ames Research Center
MS 239-15
Moffett Field, CA. 94035

ABSTRACT

Simulated docking maneuvers were performed to assess the effect of initial velocity on docking failure rate, mission duration, and Δv (fuel consumption). Subjects performed simulated docking maneuvers of an orbital maneuvering vehicle (OMV) to a space station. The effect of the removal of the range and rate displays (simulating a ranging instrumentation failure) was also examined. Naive subjects were capable of achieving a high success rate in performing simulated docking maneuvers without extensive training. Failure rate was a function of individual differences; there was no treatment effect on failure rate. The amount of time subjects reserved for final approach increased with starting velocity. Piloting of docking maneuvers was not significantly affected in any way by the removal of range and rate displays. Radial impulse was significant both by subject and by treatment. NASA's "0.1% rule", dictating an approach rate no greater than 0.1% of the range, is seen to be overly conservative for nominal docking missions.

INTRODUCTION

The relative motion of Orbital Maneuvering Vehicles (OMVs) with respect to a space station is very difficult to visualize because of non-linearities in the governing equations of motion. For example, purely posigrade thrusts ultimately yield upward and then backward motion with respect to a "stationary" target such as a space station. Conversely, retrograde burns ultimately produce downward and forward motion. These paths are curvilinear and it is possible to "bounce" one's way around an orbit. While these burns also alter the original period, purely radial thrusts--which also provide fore and aft relative motion--do not affect the vehicle's orbital period. Simulation experiments are necessary to better understand pilot response to these non-intuitive phenomena.

BACKGROUND

Proximity operations (PROX OPS) are defined as any and all activities occurring within a one kilometer sphere of the space station. Among these are rendezvous, docking, and rescue maneuvers. Simulations (and actual performance) of rendezvous and docking maneuvers began in the Gemini era in preparation for the two docking maneuvers required for a manned mission to the moon. These operations were performed slowly to increase safety margins in the event of an incorrect burn and to minimize fuel usage. The procedures were designed to minimize risk at the expense of time. The fact that a rendezvous may require anywhere from several hours to several days to perform[9] or that a docking may take as long as several hours was not a concern in the Gemini and Apollo programs. Mission durations were set to be long enough to accomplish all mission objectives. If, due to a miscalculation or exceptional pilot performance, a maneuver was completed in less time than allocated, free time had essentially just been created. A job queue as such did not exist.

The space shuttle/space station environment will be highly operational to support sustained human productivity on orbit. Retrieval and re-insertion of satellites and other orbital missions will become routine. Every hour a crewperson is spending on a docking maneuver is an hour s/he is not spending on some other task. There is a financial consideration in addition to the productivity issue. To date, the monetary cost of a crewmember's time has not been given much consideration in the U.S. space program. Recently, the value of $35,000 was cited as the cost of an hour of astronaut time on-orbit.[2] NASA guidelines stipulate that a crewperson on the space station will monitor all manned approaches in addition to the pilot in the vehicle so this one hour docking maneuver is actually worth $70,000.[9] Since others on both the station and the shuttle may be passively monitoring all or part of the approach, the cost of a one hour docking maneuver may easily rise to beyond $150,000 not including the cost of fuel and other expendables. With an estimated 5-6 dockings of the shuttle to the station each year for support[12], the annual cost of docking to the space station may approach $1,000,000. The addition of OMV, orbital transfer vehicle (OTV), and ESA free flyer traffic will further increase this cost.

Current shuttle rendezvous guidelines are very conservative suggesting that a "0.1% rule" be followed. This rule dictates that the shuttle's relative closing velocity with respect to the space station (or some other target) should be limited to a value no greater than 0.1% of its range per second. For example, at a range of 1000 meters, the velocity should be 1 meter per second. After 100 seconds, the shuttle arrives at a range of 900 meters and the range rate is decreased to 0.9 m/s.[10] A docking from an initial range of one kilometer would take about one hour to complete if this guideline were followed. This is an

Similar discussion available as paper 89-0400 from the AIAA 27th Aerospace Sciences Meeting and in a future issue of the Journal of Spacecraft and Rockets.

arbitrary rule of thumb designed to afford the pilot a sufficient safety margin with which to successfully perform the maneuver. Very little rigorous human factors studies were performed to determine what the man-in-the-loop requirements or restrictions might be.

Another area in which data suggest the need for further evaluation of current docking guidelines is workload. Workload has been under intensive scrutiny in the airline industry for some time. Here, crew inactivity may be caused by cockpit automation. Related concerns include the potential for automation to reduce crew alertness or cause them to be easily distracted.[13] Certain recent airline accidents are interpreted to have been automation-induced and they may be preventable in the future by putting the human "into a more active role in the control loop".[14] Pilot-astronauts and spacecraft are analogous to airline pilots and aircraft and this workload level concern is relevant to space operations as well. Research has shown that tasks containing relatively long periods of inactivity are perceived as being high in workload.[1] Both too many and too few inputs required per unit time are potentially hazardous. Minimizing workload by mandating a slow approach velocity is not necessarily the safest approach.

Maintaining slow vehicle velocities serves to lengthen the safe range of human reaction time and reduce the likelihood of frenzied activity which may tend toward the instigation of (potentially tragic) errors. This is one of the reasons for speed limits on the nation's highways and is also why roads are not designed with a large number of tight, contiguous S-curves, which increase workload. However, too slow velocities may lead to long periods of inactivity which also may increase the incidence of accidents. To make use of the highway analogy again, curves are installed not only around obstacles but also at appropriate intervals to "awaken" the drivers whose attention may have lapsed due to the relatively mindless piloting of an automobile down a straight road for too long. In short, both too many and too few inputs required per unit time are potentially hazardous. Again, minimizing workload is not necessarily the safest approach.

Little rigorous human factors testing has been conducted to date in the area of spacecraft docking maneuvers. It has routinely been assumed that in spaceflight--and other activities requiring manual control--that the human is such a marvelous machine that it can adapt to any operational environment. With the awareness of the aforementioned concerns of error incidence, time and productivity issues, and cost, it is time to perform some experiments with which to better understand these considerations in the hopes of alleviating or minimizing potential problems.

The author has been involved with human factors investigations of proximity operations for a number of years. Previous work in a spacecraft flight simulator with naive test subjects yielded remarkable results concerning the ability to dock a manned OMV to the space station. While NASA policy stipulates the requirement of 1,000 hours of high performance jet pilot experience for its pilots, none of the author's test subjects had any jet pilot experience. Nevertheless, successful dockings were achieved in less than 4% of the time that would have been obtained had the 0.1% rule been followed. Mission costs were cut substantially as well. In essence, students without the benefit of jet pilot experience or (except for one subject) any kind of flight training, instruction in orbital mechanics effects (except for another subject), or any form of NASA training, achieved a high success rate "speeding" toward the space station in simulated dockings. Indeed, one simulator study claims that almost anyone can perform a successful docking maneuver with great precision with documented cases including data from secretaries and experienced pilots.[8]

This study was a partial replication of the earlier one.[2] While the two studies were similar in that they both sought to examine the effect of different docking velocities on failure rate, fuel consumption, and mission duration, there were enough fundamental differences in methodology, hardware, and software to justify a new study. These differences include the addition of an accurate star field in the background of the current study, a different thruster control system, different environments, different computers and displays, and different points of view. (The earlier study consisted of a pilot flying his craft toward the station while the current study involves remotely controlled docking.)

EXPERIMENTAL METHODS AND APPARATUS

This experiment was conducted in the Space Station Proximity Operations (PROX OPS) Simulator at NASA/Ames Research Center. The simulator primarily consists of one 3-degree-of-freedom hand controller and three "windows" on which the computer-generated imagery is presented. Buttons on the hand controller are used to select the thruster acceleration values for each axis among choices of 0.01, 0.1 and 1.0 m/s. Detailed descriptions of the simulator are available elsewhere.[4,5,6,7]

Test subjects were required to "fly" simulated remote docking maneuvers of an OMV to a space station in a 270 nautical mile orbit beginning from an initial range of 304.8 m (1,000 ft) on the -V-bar (along the velocity vector in the minus direction). A repeated measures design was used with ten missions flown at each of five initial velocities: 0.3, 3.0, 5.0, 7.0, and 9.0 m/s. From this direction, orbital mechanics effects cause the vehicle to rise. The subjects were instructed to counteract this tendency by making downward burns to accomplish a successful docking. The order that these velocities were presented was randomized and was different for each subject. Subjects were requested to resist boredom at the slowest velocity and were prohibited from accelerating to decrease the mission duration. In addition, each subject also performed ten attempts without the benefit of operational range and rate displays. These trials were performed last (at an initial velocity of 3 m/s) as they were presumed to be the most difficult and the trials with the displays would serve as practice.

Each subject was issued a training manual for perusal prior to experimentation. Training consisted of performing ten successful dockings with an initial velocity of 3.0 m/s. Once ten successful dockings were achieved, training was considered complete and data collection began.

Certain range and rate conditions had to be satisfied for a docking attempt to be considered successful. These were: a forward range of 2.0 m with an approach velocity no greater than 0.15 m/s, and up/down and left/right ranges and rates with absolute values that did not exceed 0.23 m and 0.06 m/s respectively. These values were derived from the proceedings of a NASA workshop on rendezvous and docking and were believed to be the most recent.[11]

RESULTS

Eight male subjects were tested for approximately six hours each. Time considerations prevented some of the subjects from completing all ten of the runs at 0.3 m/s but no subject performed fewer than eight. Qualitative data in the form of comments by the subjects in addition to quantitative data concerning mission duration, fuel consumption, and error incidence were recorded. Only one of the subjects had previous pilot experience (subject three).

The values for per cent unsuccessful were determined by dividing the number of failed missions for each subject by the number attempted and multiplying by 100. These values were then summed which is why the total sometimes exceeded 100%. (See Figure 1.) While most of the unsuccessful missions were not tragic in nature, for experimental purposes, "unsuccessful" was operationally defined as not satisfying the terminal range and rate conditions mentioned earlier.

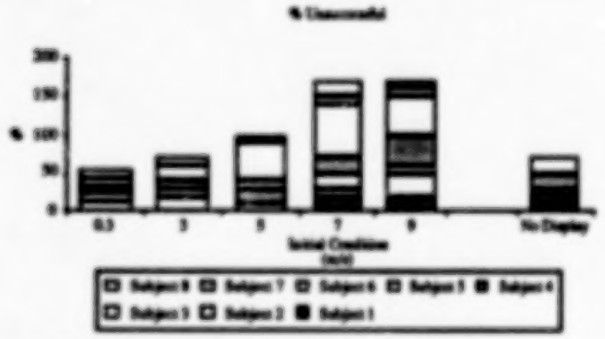


Figure 1--Per Cent Unsuccessful by Initial Condition

The data were analyzed across initial conditions and across subjects. By initial condition, the values for per cent unsuccessful ranged from 6.70% (5 "accidents") at 0.3 m/s to 21.25% (17 unsuccessful) at both 7 and 9 m/s. The corresponding values by subject were 1.68% (1 failure) for Subject 8 to 30.0% (18 failures) for Subject 5. The average failure rate by subject or initial condition was 13.2%. A two-way analysis of variance (ANOVA) was performed on these data with subject and initial condition (velocity) as factors. Only the between subject data were significant with an F-ratio of $F(7,35)=3.38$ ($p=.007$).

Median scores for mission duration and total impulse were plotted as functions of initial condition. The units for these measurements are seconds (s), and meters per second (m/s). Meters per second are the units for Δv which is the change in velocity imparted to the vehicle. By using the total impulse as the value for fuel usage, the mass of the vehicle becomes immaterial and the fuel consumption can be scaled for a vehicle of any mass. These data appear in figures 2-3.

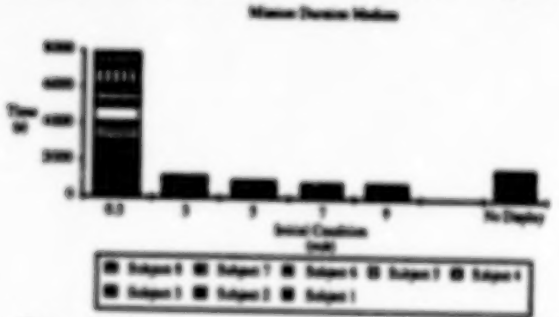


Figure 2--Mission Duration Medians by Initial Condition

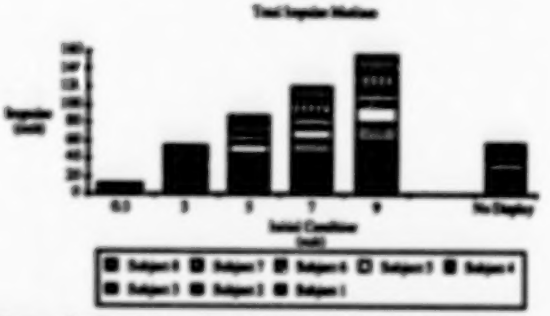


Figure 3--Total Impulse Medians by Initial Condition

Since mission duration and total impulse were heavily influenced by the initial velocity of the vehicle, the data for these parameters are more important for mission operations considerations than for human performance analyses. Consequently, mission duration and total impulse were "normalized" by subtracting out appropriate reference values. The reference value for mission duration was computed by dividing the initial range by the initial velocity. This provides a theoretical minimum time for a linear, one-dimensional system (which does not characterize the orbital environment) with impulsive start and stop. The parameter thus obtained is termed "reserve time" as this is the time the subject reserves for himself in order to successfully dock.

In a similar fashion, the starting impulse and the impulses used to decelerate the vehicle were subtracted from the total impulse to arrive at the value used to maintain the OMV near the V-bar. (Since the OMV rarely came to a full stop, the vehicle was assumed to have a residual velocity of 0.1 m/s for this calculation.) This derived parameter was termed "radial impulse" as this was the sum of the radial impulses used to achieve a successful docking.

Reserve time medians averaged over all subjects ranged from a low of -22.2 s at 0.3 m/s to a high of 108.4 s with the No Display trials. (A negative reserve time was caused by reducing the altitude of the OMV for so long that its orbital period was significantly shorter from that of the station and it gained some forward velocity.) Across treatments, the averages ranged from 23.8 s (Subject 4) to 114 s (Subject 1). For the reserve time medians, the omnibus F test produced ratios of $F(5,35)=6.73$ ($p<.001$) and $F(7,35)=2.23$ ($p=.055$) for the between treatment and between subject data respectively. (See Figure 4.)

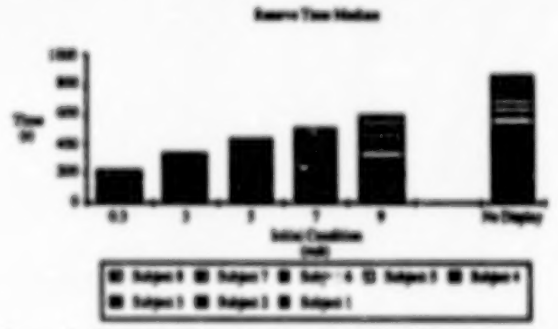


Figure 4--Reserve Time Medians by Initial Condition

The F-ratios for radial impulse medians were $F(5,35)=2.81$ ($p=.031$) and $F(7,35)=6.68$ ($p<.001$) for between treatment and between subject analyses. Averages of medians across subjects ranged from 1.02 m/s at 0.3 m/s to 1.77 m/s at 9

m/s. Across treatments, the values varied from 0.75 m/s for subject three to 2.1 m/s for subject five with an average of 1.26 m/s. (See Figure 5.)

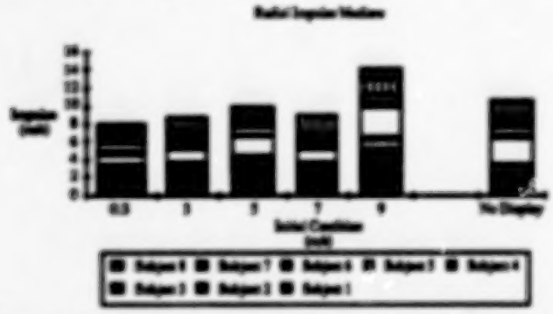


Figure 5--Radial Impulse Medians by Initial Condition

Two-way ANOVAs were also conducted for the three parameters using only the 3 m/s and No Display (also 3 m/s) medians to determine the effect of the removal of the displays. None of these F ratios proved significant.

DISCUSSION--ANALYSIS

Before discussing the results in any detail, it must be emphasized that none of the subjects had any background or experience with orbital mechanics effects or any high performance jet flight training. The data show how well naive subjects can do without either of these skills. Also, none of the pilots had access to any orbital trajectory planning device to assist them with their docking missions. (This is being planned for a future study.) It is expected that NASA pilots with these skills, and more, could easily surpass the best mission duration, fuel consumption, and success rate values achieved here. Subjects and NASA pilots alike can be trained to virtually any desired design point for any, or all, of the three parameters before they are considered competent to perform an actual docking.

The qualitative comments clearly indicated extreme dislike for the trials at the slowest velocity and preference for any of the other treatments, especially the runs without the range and rate displays. However, the anxiety associated with the performance of an actual space mission may alleviate some of the boredom and help time "fly".

The data obtained for the unsuccessful missions, categorized by initial condition and by test subject, show that while the incidence of accidents appears to increase with initial velocity, this trend was not statistically significant. Only the subject effect was significant. Subjects had different risk profiles; some were risk prone and some were risk averse. In actuality, docking velocities would not be chosen because of their associated success rate in simulations. Rather, the pilots would be chosen by their established success rate in simulated maneuvers. Unlike other vehicles such as aircraft or automobiles where the landing scheme or speed limit must be designed to safely accommodate the worst pilots (drivers), for spacecraft and space missions, "the simulator defines the user population rather than vice versa." A velocity would not be selected because it induced the lowest failure rate averaged over all subjects, rather, those who performed the best in the simulated mission would be chosen to perform the actual mission.[4]

The analysis of variance performed on reserve time medians showed that reserve time increased with initial velocity without any statistical significance among test subjects. While mission duration varied inversely with initial velocity, reserve time increased monotonically with initial velocity with the no display runs performed at 3 m/s requiring more reserve time than the runs at 9 m/s. This effect was mostly due to the equations of motion governing orbital flight. By traveling at a different altitude from the station, forward velocity was obtained without the use of fuel. During the longer (slower) missions, orbital mechanics effects had more time to work to the subjects' advantage allowing more velocity to be accumulated and consequently reducing the time the pilot reserved to successfully accomplish the mission.

The radial impulse data were significant both by subject and by treatment. This indicates that not only were different amounts of fuel required to compensate for a non-linear environment depending upon the initial velocity, but that some subjects were significantly more fuel efficient than others when it came to applying these radial burns.

The fact that removal of the range and rate displays at 3 m/s did not significantly affect the data for any of the parameters is very important. This indicates that such displays, while probably psychologically comforting, were redundant in a real-life situation when combined with the visual image of the approaching vehicle and were unnecessary for nominal dockings. They did not help the test subjects perform more rapid, more fuel efficient, or safer dockings. They would most likely be more useful in anomalous situations. These range and rate data would be produced by a sophisticated and expensive ranging system which the current space shuttle does not possess and it is deemed too expensive to retrofit the shuttle to include it. This lends support to the view that installing such a system on the shuttle is not worth the time, money, or effort since these displays apparently do not produce any significant benefit in simulated nominal missions. The OMV is expected to have such a ranging system (which is one of the reasons it was used as the test vehicle in this study) and these data would be required for an automatic system to function as well.

CONCLUSIONS

- The amount of time pilots reserved for final approach increased with starting velocity. The slower forward velocities allowed more time for the orbital mechanics effects to play a role and were thus used to the pilot's advantage in gaining forward velocity for free.
- Performance of simulated remote piloting of docking maneuvers was not significantly affected in any way by the removal of the range and rate displays.
- The initial condition significantly affected the subjects' use of reserve time and radial impulse.
- The "0.1% Rule" for docking is overly conservative from a human performance point of view.

REFERENCES

[1] Awe, Cynthia A., Boredom and Workload: The Relationship Between Task Difficulty, Duration, and Rate, *Proceedings of the 23rd Annual Conference on Manual Control*, Cambridge, MA, June, 1988.

[2] Ballard, R., Implications of Basic Biological Research for Future Long-Duration Missions, *Aerospace Medical Association 59th Annual Scientific Meeting*, New Orleans, LA, May, 1988

[3] Brody, Adam Randall, Spacecraft Flight Simulation: A Human Factors Investigation Into the Man-Machine Interface Between an Astronaut and a Spacecraft Performing Docking Maneuvers and Other Proximity Operations, NASA-CR-177502, NASA/Ames Research Center, Moffett Field, CA, September, 1988.

[4] Brody, Adam R., Modifications to the NASA/Ames Space Station Proximity Operations (PROX OPS) Simulator, NASA-CR-177510, NASA/Ames Research Center, Moffett Field, CA, October, 1988.

[5] Haines, Richard F., Design and Development of a Space Station Proximity Operations Research and Development Mockup, SAE Technical Paper 861785, 1986.

[6] Haines, R. F., and Hamilton, J., Space Vehicle Size Influences Judgements of Radial Approach, *Aerospace Medical Association 59th Annual Scientific Meeting*, New Orleans, LA, May, 1988.

[7] Lee, E., and A. Wu, Space Station Proximity Operations Workstation Docking Simulation, Sterling Software TN-87-7104-519-13, Palo Alto, CA, March, 1987.

[8] Levin, E., and Ward, J., Manned Control of Orbital Rendezvous, *Proceedings of the Manned Space Stations Symposium*, Los Angeles, CA, April, 1960.

[9] NASA/Lyndon B. Johnson Space Center, Space Station Reference Configuration Description, *Systems Engineering and Integration Space Station Program Office*, NASA/Johnson Space Center, Houston, TX, 1984.

[10] NASA/Lyndon B. Johnson Space Center, Rendezvous/Proximity Operations Workbook, RNDZ 2102, NASA/Johnson Space Center, Houston, TX, 1985.

[11] NASA/Lyndon B. Johnson Space Center, Proceedings of the Rendezvous and Proximity Operations Workshop, NASA/Johnson Space Center, Houston, TX, February, 1985.

[12] Pool, S. L., Extension of Crew Stay Time, *Aerospace Medical Association 59th Annual Scientific Meeting*, New Orleans, LA, May, 1988.

[13] Wiener, Earl L., Beyond the Sterile Cockpit, Human Factors, 27(1), 1985.

[14] Wiener, Earl L., Computers in the Cockpit: But What About the Pilots?, SAE Technical Paper 831546, 1983.

A Helmet Mounted Display to Adapt the Telerobotic Environment to Human Vision

Gregory Tharp, Andrew Liu, Hiromi Yamashita, Lawrence Stark
Brenda Wong, Jurgen Dee

Telerobotics Unit, 481 Minor Hall,
University of California, Berkeley, CA 94720

Abstract

A Helmet Mounted Display system has been developed. It provides the capability to display stereo images with the viewpoint tied to subjects head orientation. This type of display might be useful in a telerobotic environment provided the correct operating parameters are known. The effects of update frequency were tested using a 3D tracking task. The effects of blur were tested using both tracking and pick-and-place tasks. For both we found that operator performance can be degraded if the correct parameters are not used. We are also using the display to explore the use of head movements as part of gaze as subjects search their visual field for target objects.

Introduction

Stereo displays are now commonly used as a visual interface for telerobotic systems because they provide better telepresence than monoscopic displays [1],[3],[13]. The hardware has improved considerably since stereo displays have been used, but our knowledge of the perceptual aspects of using stereo display systems remains very limited. The video images from a remote site could be degraded by a number of factors such as blur, noise, contrast or illumination changes that could differentially affect performance with stereo and non-stereo displays. The effects of these degradations on performance has not been closely examined except in the case of contrast [4]. Without this knowledge, stereo displays might be used in inappropriate situations, i.e. where an unnecessary increase in performance is offset by increases in cost, transmission bandwidth or system complexity [5].

In order to test the usefulness of our Helmet Mounted Display system, we decided to explore how changing the parameters of the display affected the performance of some representative telerobotic tasks. Previous studies have shown that increasing the apparent interocular distance of the display from zero, decreases mean completion time for a pick and place task.[1] Here we explore the effects of different update frequencies and blur on performance.

The Helmet Mounted Display

Helmet mounted displays were designed as a more comfortable interface that provides the human operator with telepresence, i.e. the feeling that he/she is in the remote environment and controlling the manipulator directly. The helmet is equipped with a stereo display and can detect the operator's head motions in order to move the remote stereo cameras in a similar fashion. Our experiments simulated the remote environment with computer graphics and standard stereoscopic display formulas [1].

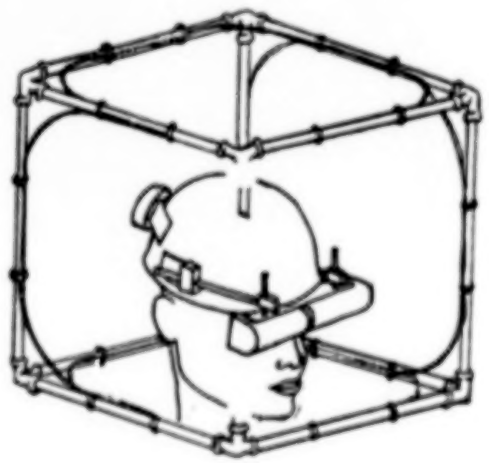


Figure 1. Helmet Mounted Display This system provides a stereo view through two video camera viewfinders, one for each eye. Head orientation is measured with the Helmholtz coil system. (outer coils)

Two degrees-of-freedom are detected by the head orientation sensing device, horizontal and vertical rotations. The sensing system is comprised of two magnetic Helmholtz coils with normal magnetic fields and a sensing coil mounted on the inside top of the helmet. Each field coil generates a different sine wave magnetic flux (50kHz for horizontal, 75kHz for vertical). The induced flux in the sensing coil is then amplified and separated into the respective components. The magnitude of each component is dependent on the head orientation with respect to the corresponding field coil [2].

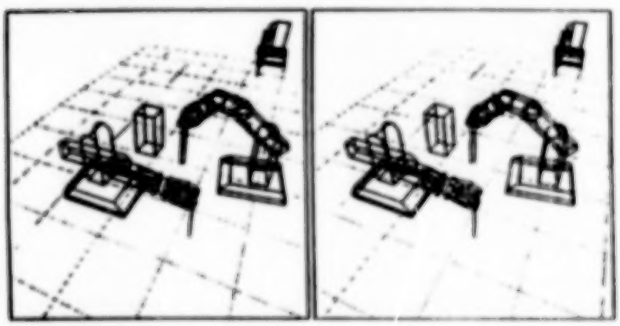


Figure 2. Robot Manipuation Task. Stereo images and head coordinated views give operators a sense of telepresense. The relationship between robots and objects might be more easily seen in such a display.

The stereo display is comprised of two 1-inch viewfinders (Sony VF-208) mounted on the helmet by means of a 5 degree-of-freedom frame. The frame has three orthogonal slidings and two rotations. Each viewfinder also has a converging lens which forms the virtual image of the stereo pair 25cm behind the actual display screen. The sixfold magnification of the lens makes the display screen look like a 6-inch screen (33 degrees wide). When adjusted appropriately, most observers can fuse the two images into a single three-dimensional image. A 0.5 kg counterweight was fastened to the back of the helmet to counterbalance the viewfinders.

Update Frequency Requirements

One important goal of using a HMD system is to provide the user with a stronger sense of telepresence from the enforced correlation of head movements and displayed view direction. If the scene is not updated rapidly enough this correlation (called Space Constancy in the realm of natural human gaze movements and psychophysics [15]) might be lost, but in a telerobotic setting, limitations in hardware or in signal bandwidth might limit the update speed. We suspect that there is some band of update frequencies above which there is no noticeable change in the display quality but below which operator performance falls to unacceptable levels. It is this range where performance is affected by update frequency that we explore here.

We tested the effects of update frequency by using a 3D tracking task. Figure 3 shows a sample display from the experiment. The subjects viewed a virtual space from above the edge of a ground grid. In the space was a box target and a spherical cursor. The cursor was controlled through two joysticks. The subjects were asked to keep the sphere in the box. The target moved in a pseudorandom pattern in three dimensions. The target trajectory was a sum of 11 sinusoids with frequencies from 0.007 Hz to 5.8 Hz. The amplitudes simulated a low pass filter with a corner frequency of about 0.08 Hz. The field of view of the scene extended to 60 degrees so that when the target was at the extremes of its motion, head movements were performed to keep the target in the visual field. Under these conditions each subject carried out the task for 10 update frequencies ranging from 15 Hz to 0.5 Hz presented in a random order.

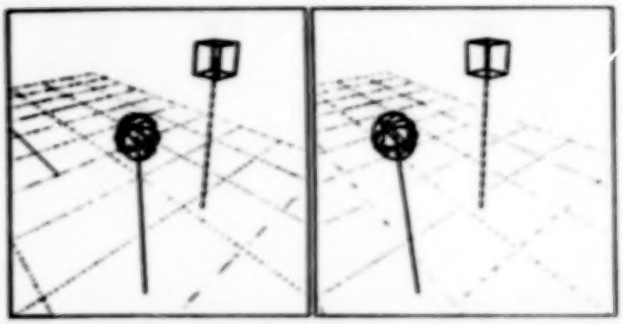


Figure 3. 3D Tracking Tasks. The subject tries to keep the ball inside the box that continually moves in a pseudorandom pattern. The ball is controlled by two joysticks.

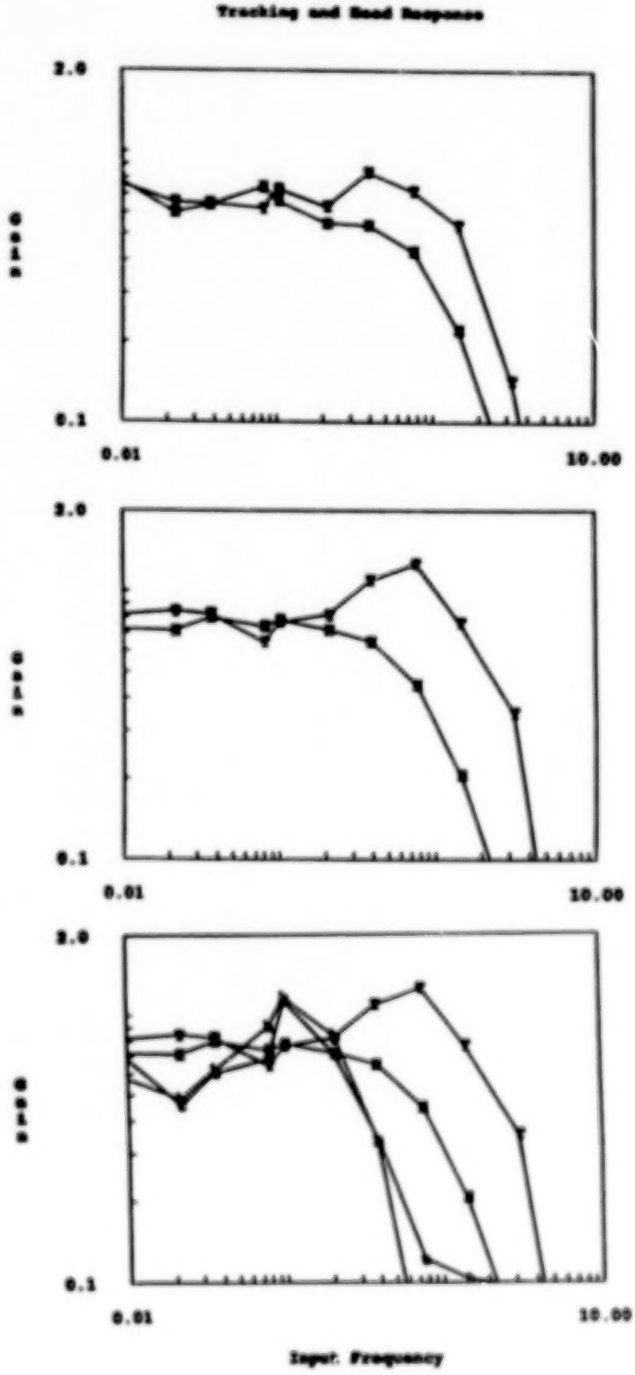


Figure 5. Tracking Response. Two subjects (top and middle panels) show that the tracking [T] has a higher bandwidth than the head movements [H] at the highest update rate. At the lowest update rate (bottom panel) tracking [t] and head movements [h] have the same bandwidth.

Due to the high frequency content of the input signal, the normalized rms error is high even for the fastest update frequency. We normalized the rms error to the situation where there was no operator input. Therefore, error values greater than one indicate tracking worse than doing nothing at all. This situation occurs for the lowest group of update frequencies that we tested. Figure 4 shows the rms error for all the update rates tested.

We measured both the cursor position and the head movements even though the subjects were not told to track the target with their head. Figure 5 (top and middle) shows the response of two subjects at the highest update frequency for both tracking and head movements. It shows that the bandwidth of the head movements is smaller than that of the tracking. This is not unexpected since the subjects were only using head movements to assist in the tracking. However, the bottom panel, which also shows the response at the lowest update frequency, indicates that manual tracking degrades to the same bandwidth as head movements. Even so, figure 6 shows that at a particular input frequency, around the corner frequency of the tracking at the highest update rate, the manual tracking is better than head movements throughout the experiment. In general we discovered the not surprising result that if the image doesn't change with head movement, subjects tend not to move their heads. But, they seem to try to track the target even after the update rate slows considerably.

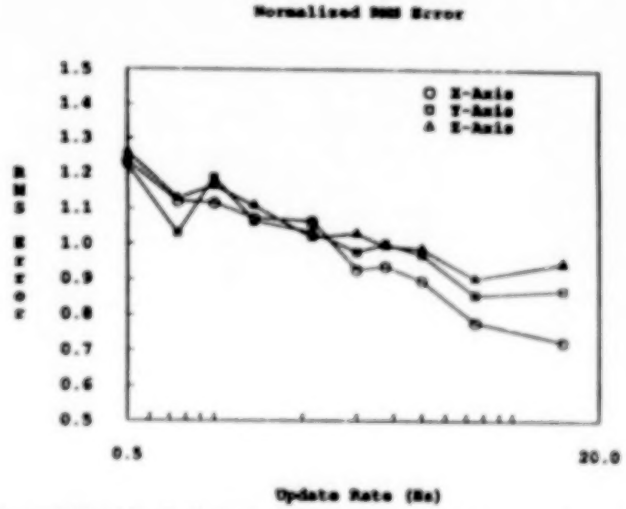


Figure 4. Tracking Performance. The rms error is normalized to the case of no response. As the update frequency falls the rms error rises; at the lowest frequencies the tracking is worse than no response.

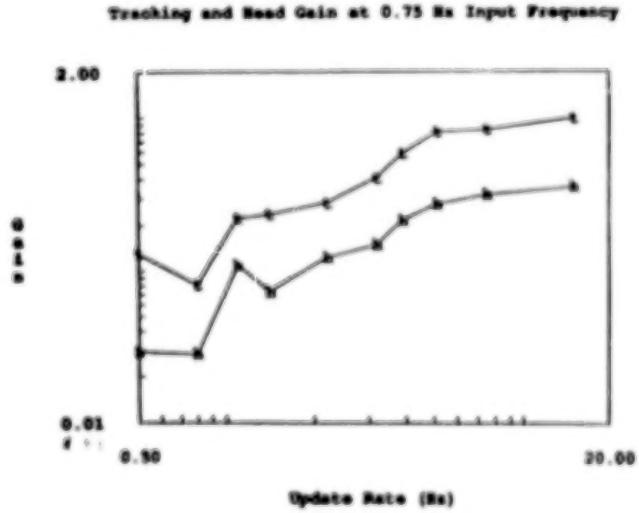


Figure 6. Tracking Gains. The gains of the tracking and head movements for an input frequency of 0.75Hz across all update frequencies shows how the gains fall as update frequency goes down. It also shows that tracking is more accurate than head movements, this is because the subjects were not asked to track the target with their heads.

The Effects of Blur on Task Performance

Psychophysical experiments can provide a basis for predicting performance changes under these degrading conditions. For example, experiments studying the effect of blur on human stereoacuity have shown that target blur reduces stereoacuity. Most of these experiments used blur induced with spectacle lenses and measured the stereoacuity with a depth-discrimination test such as the Howard-Dolman test [6]-[9]. Blur caused by spatial frequency filtering [8] has also been shown to cause similar degradations in stereoacuity. In addition, it was found that blur of only one eye resulted in a greater deficit than equal amounts of blur in both eyes. Figure 7 (top) shows some of the results from Lit's and Westheimer and McKee's experiments with monocular and binocular spectacle blur. The data for binocular blur is shown by the solid plot symbols. Stereo displays differ from monoscopic displays in that they also use the disparity cue to display depth. Therefore, we conclude that stereoacuity, or the ability to discriminate two points in depth, is important in telemanipulation. Furthermore, we would expect operator performance to decrease as the one or both of the images were increasingly blurred.

To confirm this hypothesis, we examined the effect of visual target blur on the performance of simulated telerobotic tasks. Using the helmet mounted display described previously, two generic telerobotic tasks were simulated, a pick-and-place task and three-axis tracking. Head movements were not needed to view the entire scene in these experiments. Also, the binocular disparity in the scene provided the only cue to the depth of the targets in space. The pick-and-place task required the operator to grasp objects that were floating within the workspace and put them in a box on the floor. Performance was measured by the average time to complete the task for 20 objects. In the three-axis tracking task, the operator followed a randomly moving target as closely as possible with a cursor controlled by two displacement joysticks. Performance was measured by the rms error between the target and the cursor. The error was normalized by the error value associated with no movement of the operator's cursor.

These tasks were performed under bioptic (binocular without disparity), ideal stereoscopic and blurred stereoscopic viewing conditions. The simulated scenes were drawn with lines and dots in order to reduce the number of computations needed to display and update the scene. Monocular target blur was induced by spectacle lenses and used in all the trials. This was done to ensure that a reduction in stereoacuity and performance occurred. Since disparity was the only depth cue present in the scene, performance was expected to deteriorate in a similar fashion to stereoacuity. Our results are directly comparable to the psychophysical results since the method for inducing blur is the same.

In Figure 7 (middle and bottom) the results show that performance of the pick-and-place and three-axis tracking tasks decreased when 2 diopters of monocular blur was present. The performance fell to the level of performance for the bioptic(B) viewing condition. Psychophysical results show that stereoacuity becomes 3-15 times worse with the addition of 2 diopters blur. This suggests that moderate reductions in stereoacuity can be enough to eliminate the advantage of using stereo displays. This helps explain how early experiments comparing stereoscopic and monoscopic displays did not find any performance differences.[14] It is possible that the stereo displays were not carefully adjusted such that the images seen by the operator was blurry. As a result, their stereoacuity was degraded to the point that the stereo display was not useful.

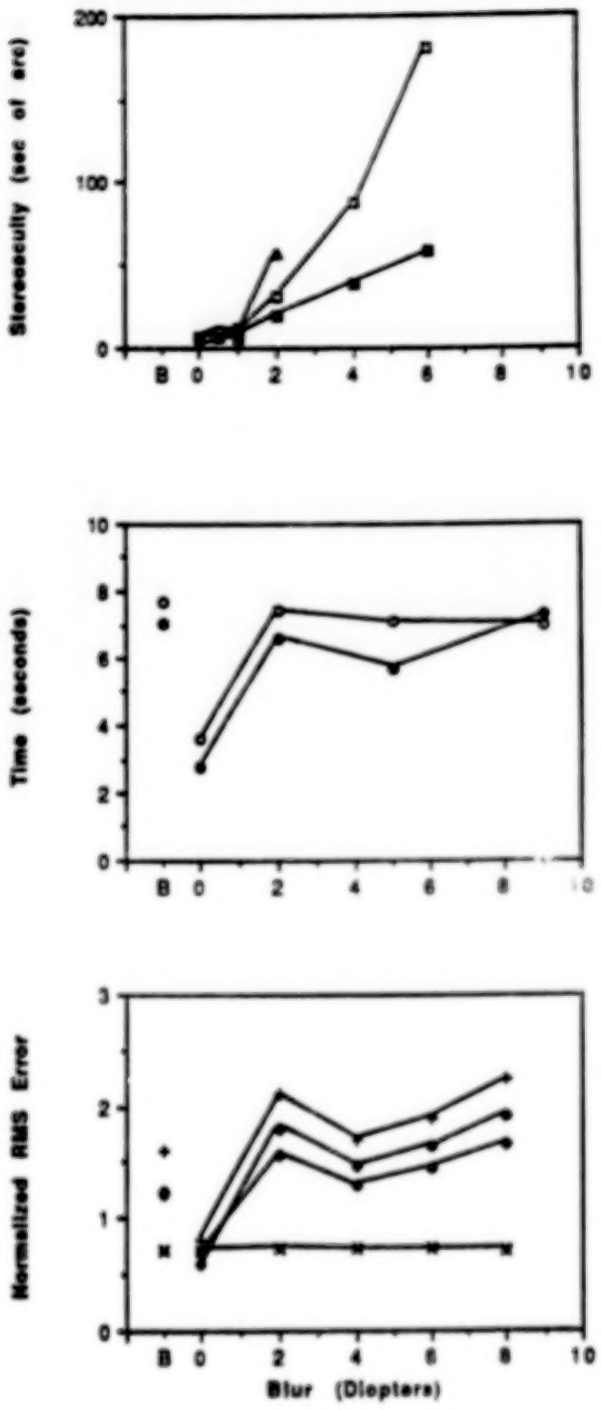


Figure 7. Blur Effects. The effect of spectacle blur on stereoacuity and task performance. Top: Data from Lit's (square) and Westheimer and McKee's (triangle) experiments. Middle: Mean completion time for the pick-and-place task versus blur. The circle symbol is for an average of seven trials, solid circle is for one trial. Bottom: Error in tracking vs. blur. We examined the error along the *x* [horizontal](cross), *y* [vertical](plus), *z* [zoom](diamond) axes and also in three dimensions (solid)

Previous Blur Studies

Previous studies of blur as the stimulus to accommodation have been carried out by Dr. Stephen Phillips in our laboratory. Two aspects of the blur signal are of particular interest: — (i) the eccentricity of the stimulus with respect to the retinal fovea, and (ii) the spatial frequencies of the target before it is becomes the blurred retinal image.

Two examples of the effect of eccentricity on the static and dynamic tracking ability of the accommodative system are shown in Figures 8 and 9. It is clear that the ability of a defocused peripheral target evoking an accommodative response drops off rapidly outside the fovea; these results correlate with the decreased visual acuity. In normal viewing conditions vergence-accommodative synkinesis controls accommodation whenever a focusable target is more than several degrees off the line-of-sight.

Spatial frequencies between 1.0 c/deg and 25 c/deg (at a 0.63 contrast ratio, Cm) are effective as shown in the experiment of Figure 10. Here, several subjects are focusing at a 5.5 D and 0 D distant target through a range of spatial frequencies. When the spatial frequency go beyond the feasible range the subjects drift back to their bias set level (generally about 1.0 diopter, but for one of our subjects to about 2.0 diopter due to instrumental myopia). A summary figure , Figure 11, shows the envelope of responses to adequate and inadequate spatial frequency targets over a range of accommodative stimuli.

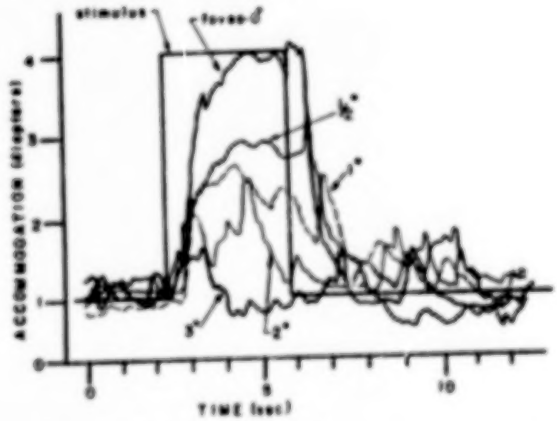


Figure 8. Accommodation response as a function of eccentricity. A 4 s duration step stimulus at 4 D produced an excellent response at 0 deg eccentricity, but even a 1/2 deg eccentricity greatly reduced response. Response at 2 deg is already likely only a searching stimulated by the subject noticing a change in light distribution. (Phillips, S. Thesis, 1974, University of California, Berkeley).

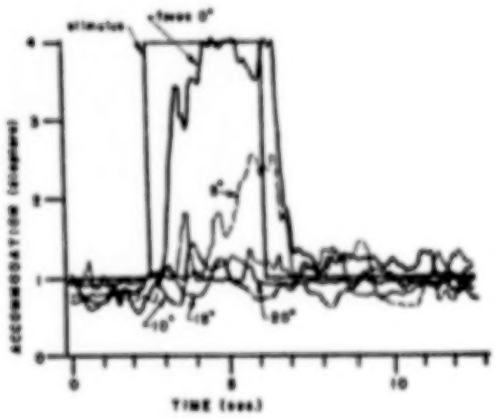


Figure 9. Accommodation Responses at Higher Eccentricities. In another subject, with excellent response at 0 deg eccentricity, little on no responses occurs at moderate eccentricities. (Phillips, S. Thesis, 1974, University of California, Berkeley).

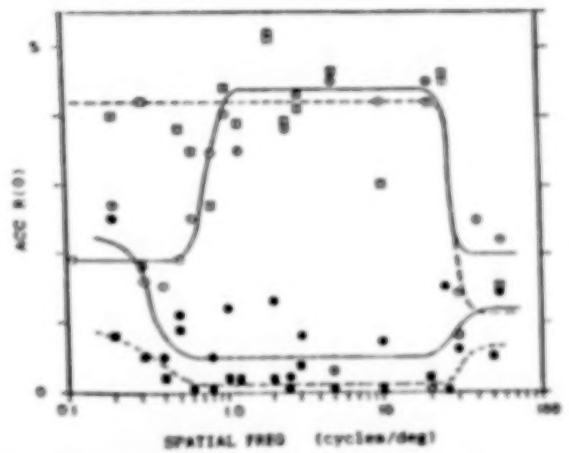


Figure 10. Range of spatial frequencies serving as adequate stimulus to accommodation. Accommodative stimuli, containing frequencies from 1.0 c/deg to 25 c/deg were adequate as shown in responses to 5 D and 0 D from a stimulus level of 1 D, for two subjects. Bias set level for one subject was 2 D. (Phillips, S. Thesis, 1974, University of California, Berkeley).

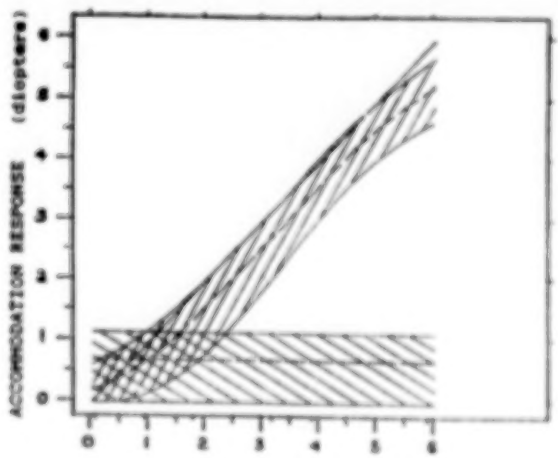


Figure 11. Envelopes of responses to adequate and inadequate stimuli. Accommodative stimuli containing spatial frequencies from 1.0 c/deg to 25 c/deg were able to drive accommodation accurately over the range of diopters from 0 to 6 D (upper envelope). Stimuli containing only frequencies outside that range were inadequate (lower envelope). (Phillips, S. Thesis, 1974, University of California, Berkeley).

Visual Search

Visual scanning behavior, or how humans search their visual field, can be investigated through the use of the helmet-mounted display. These types of investigations of visual search would be useful in the development of electro-optical detection devices and manned systems designs. Much research has been done concerning visual scanning behavior using eye-movement data; however, head movements are also an interactive part of the visual scanning process, and is an area worth investigating. With the helmet-mounted display, viewing monitor windows limit the visual field (the area of extent of physical space visible to an eye in a given position) to a measurable amount, while the field of view (the extent of the object plane visible through, or imaged by an optical instrument) is digitally controllable. Thus, the subject is induced to use head movements with limited eye movements to search a given scene in a simulated field.

Figure 12 shows an example of head-movement data superimposed on the searched scene. A 3120 Silicon Graphics Iris was used to create and display this scene of robots on a simple background. Through the HMD monitors, only a windowed portion of the scene was visible to the subject at a time (see Figure 13). As the subject turned his/her head, a perspective projection of the scene was displayed at an appropriate angle. The resulting effect was one of looking at a large two-dimensional wall. Figure 14 shows the actual display appearing on the HMD monitors while the subject viewed a central portion of the scene in fig 12. Here, the simulated dimensions used were of a 10 ¼' x 7 ¾' wall viewed from a distance of 2'. A 60-degree head rotation was required to scan the entire scene.

Obviously, there are many factors which can affect visual scanning behavior. These include display resolution; number, orientation, and size of targets; amount of clutter (non-target objects resembling the target); field of view; noise; and peripheral visual acuity. Various eye movement studies on visual search have been conducted investigating the effect of the visual lobe area on visual search [10]-[12]. The visual lobe area is defined as the area surrounding the central fixation point from which information can be obtained in a single glance. As the complexity of the background in which the target is embedded is increased, the size of the visual lobe area decreases. Ultimately, the visual lobe area is a major factor influencing search time. This would be an interesting subject to investigate further using head-movement data.

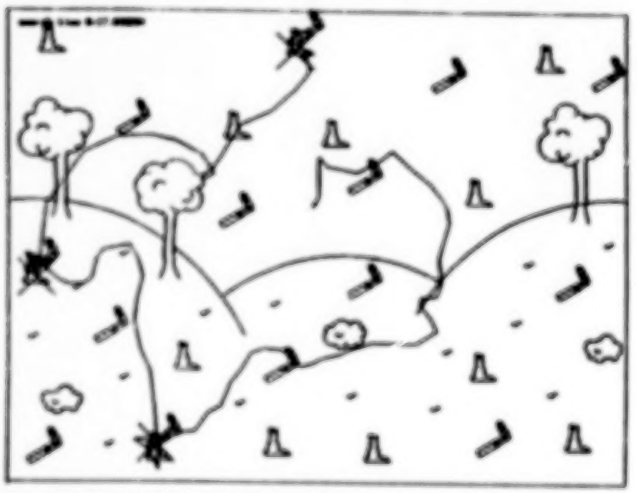


Figure 12. Head Movement Driven Visual Search. A typical search path (heavy line) with X's marking acquired targets, complete robots, in a field with clutter (partial robots) demonstrates that subjects do not make systematic searches.

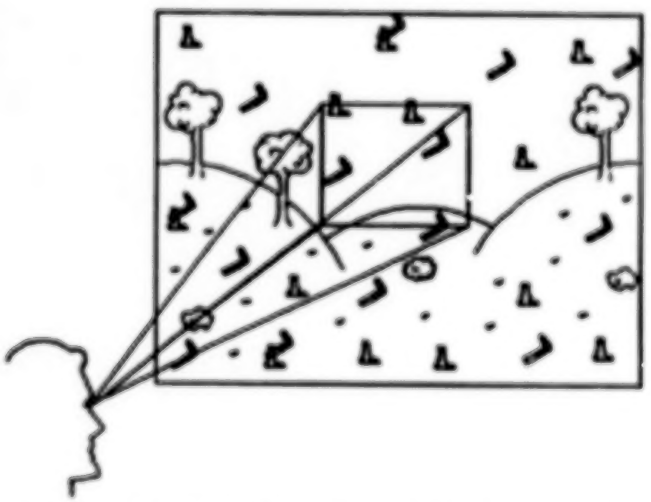


Figure 13. Visible Window Control by Head Direction. To study visual search, we limited the solid window angle on a large display solid angle. This forced subjects to search the space with a combination of head and eye movements.



Figure 14. The Visual Search Display. The HMD window provides only a limited view of the scene. In the figure the left image is inverted as in the actual display because the left viewfinder of the HMD is upsidedown.

Acknowledgements

We would like to acknowledge the work of Drs. Huy Ngo and Neil Crawford who provided much support at the inception of the visual search experiments. We are also grateful for support and encouragement from NASA-Ames (Dr. S. Ellis), JPL (Dr. A. Bejczy), TRAC-White Sands (Drs. D. Collier and E. Payan) and Varilux-Essilor (Dr. G. Obrecht).

References

[1] W. S. Kim, S. R. Ellis, M. E. Tyler, B. Hannaford and L. W. Stark, "Quantitative evaluation of perspective and stereoscopic displays in three-axis tracking tasks," *IEEE Trans. Syst., Man, Cybern.*, vol. 17, pp. 61-72, 1987.

[2] M. K. Duffy, "A Head Monitor System Using the Search Coil Method," Master's Thesis, Dept. of Elec. Engr., Univ. of California, Berkeley, 1985.

[3] W. S. Kim, F. Tendick and L. W. Stark, "Visual Enhancements in Pick-and-Place Tasks: Human Operators Controlling a Cylindrical Manipulator," *IEEE J. Rob. Autom.*, vol. 3, pp. 418-425, 1987.

[4] R. L. Pepper, D. C. Smith and R. E. Cole, "Stereo TV improves operator performance under degraded conditions," *Opt. Eng.*, vol. 20, pp. 579-585, 1981.

[5] P. S. Schenker, "NASA Research and Development for Space Telerobotics," *IEEE Trans. Aero. Elec. Sys.*, vol. 24, pp. 523-534, 1988.

[6] A. Lit, "Presentation of experimental data," *J. Am. Optom. Assoc.*, vol. 39, pp. 1098-1099, 1968.

[7] J. Ong and W. S. Burley, "Effect of induced anisometropia on depth perception," *Am. J. Optom. Arch. Am. Acad. Optom.*, vol. 49, pp. 333-335, 1970.

[8] G. Westheimer and S. P. McKee, "Stereoscopic acuity with defocused and spatially filtered retinal images," *J. Opt. Soc. Am.*, vol. 70, pp. 772-778, 1980.

[9] W. L. Larson and A. Lachance, "Stereoscopic Acuity with Induced Refractive Errors," *Am J. Opto. Physiol. Opt.*, vol. 60, pp. 509-513, 1983.

[10] F. L. Engel, "Visual conspicuity as an external determinant of eye movements and selective attention," Unpublished Doctoral dissertation, T.H. Eindhoven, 1976.

[11] K. F. Kraiss and A. Knaeuper, "Using Visual Lobe Area Measurements to Predict Visual Search Performance," *Human Factors*, vol. 24, pp. 673-682, 1982.

[12] H. Widdel and J. Kaster, "Eye movement measurements in the assessment and training of visual performance," In *Manned systems design, methods, equipment and applications, p. 251-270, RJ. Moraal and K. F. Kraiss (Eds.), New York: Plenum, 1981.*

[13] T.B. Sheridan, "Teleoperation, telepresence, and telerobotics: research needs for space," *In Human Factors in Automated and Robotic Space Systems*, pp. 279-285, T.B. Sherdan, D.S. Kruser, and S. Deutsch (Eds.), Washington D.C., 1987.

[14] D.E. Hudson and G. Culpit, "Stereo TV Enhancement Study, Final Technical Report," Kollsman Instrument Corp, Syosset, N.Y., February 1968.

[15] L. Stark, "Space Constancy and Corollary Discharge," *Perception and Psychophysics*, 37, pp. 272-273, 1985.

ASSESSMENT OF CONTROL STABILITY FOR A DEXTEROUS TELEOPERATOR WITH TIME DELAY

J. McConnell
Grumman Space Systems

(Paper not provided by publication date.)

HUMAN FACTORS ISSUES IN TELEROBOTIC SYSTEMS FOR SPACE STATION FREEDOM SERVICING

Thomas B. Malone, Ph.D
Kathryn E. Permenter
Carlow Associates Incorporated
8315 Lee Highway, Suite 410
Fairfax, Virginia 22031

Abstract

This paper describes requirements for Space Station Freedom servicing and defines the state-of-the-art for telerobotic system on-orbit servicing of spacecraft. The paper also identifies the projected requirements for the Space Station Flight Telerobotic Servicer (FTS). Finally, the human factors issues in telerobotic servicing are discussed. The human factors issues are basically three: the definition of the role of the human versus automation in system control; the identification of operator-device interface design requirements; and the requirements for development of an operator-machine interface simulation capability.

Space Station Freedom Servicing Modes

One of the elemental capabilities to be provided on Space Station Freedom is the on-orbit servicing of spacecraft, satellites, payloads, and Space Station elements. Servicing includes all activities associated with restoring the operational capability of a system including fault identification and diagnosis, planned maintenance, and corrective maintenance. Specific servicing tasks include inspection, fault isolation, refurbishment, removal and replacement of components, checkout, test and calibration, cleaning, tightening, adjusting, and repair.

Alternate servicing modes being considered for Space Station Freedom include astronaut extravehicular activity (EVA), telerobotic systems, and automated servicing. EVA involves servicing by an astronaut at the worksite. Telerobotic servicing involves operations controlled by a human from a location remote from the worksite. Automated servicing involves performance of servicing activities in a purely autonomous manner with no human involvement.

EVA for spacecraft servicing is a proven technology with the results of Skylab sun shield deployment and Solar Max maintenance. The advantages of EVA as a technique for servicing Space Station Freedom as compared with the other alternatives include: 1) EVA is the furthest advanced of the alternate modes in terms of validated technology; 2) EVA is the most flexible of the modes and makes fewest demands on the design of the systems being serviced; 3) the astronaut is at the worksite bringing to the task all the adaptability and flexibility inherent in human decision making, adaptive reasoning and problem solving; and 4) since interior Space Station Freedom servicing will probably be accomplished in large part by a human technician, the only functional difference between IVA and EVA servicing is the constraint of performing EVA in the pressurized suit. Disadvantages of EVA include: 1) demand on astronaut time; 2) less than optimum applicability to repetitive tasks, 3) impacts on astronaut safety; and 4) associated astronaut support requirements such as life support, workstations, lighting, tools, and astronaut training.

Telerobotic servicing has been accomplished in earth applications, such as in the hostile environments of undersea worksites and nuclear power plants. Telerobotic servicing has not been performed in space as of yet. Advantages of the telerobotic servicing mode include: 1) availability of human decision making, adaptive reasoning and problem solving without the additional hazards associated with placing the human at the worksite; 2) reduced demands for human operator time as compared with EVA; 3) ability to perform in conjunction with the other modes (combined

EVA and telerobotic servicing, and servicing tasks involving both human remote control and automated control; 4) the capability for fully repetitive actions; and 5) its ability to perform servicing tasks simultaneously with a number of other telerobotic servicing systems. Disadvantages of telerobotic servicing systems include: 1) limited capability (dexterity, reach, controllability) associated with existing telerobotic technology; 2) demands on the human operator; 3) demands placed on the design of interfaces between the telerobotic servicing system and the system being serviced; and 4) additional servicing requirements for servicing of the telerobotic system itself.

Automated servicing in terms of selecting redundant systems and automated fault detection has always been provided on manned and unmanned spacecraft. Capabilities associated with component removal and replacement, inspection, and repair in strictly an automated mode have not been developed for space missions. The advantages of automated servicing include: 1) minimal demands for crew time; 2) minimal impact on crew safety; 3) capability to perform repetitive and rote tasks continually; and 4) its inherent relationships with artificial intelligence and adaptive control. Disadvantages of automated servicing include: 1) technology advancements needed to make the mode viable; 2) loss of adaptability and flexibility in servicing operations and spacecraft design; 3) constraints placed on the design of the system to be serviced.

The investigation of The Human Role in Space (THURIS) conducted by McDonnell Douglas for Marshall Space Flight Center in 1984 categorized the alternative modes for on-orbit satellite servicing as follows:

- manual - unaided EVA with simple non-powered hand tools;
- supported - EVA servicing using supporting machinery or facilities to accomplish assigned tasks (e.g., manned maneuvering unit);
- augmented - EVA with human sensory or motor capabilities augmented (e.g., powered tools, exoskeletons, and inspection aids);
- teleoperated - use of remotely controlled sensors and actuators enabling the removal of the human from the worksite while maintaining direct manual control over the servicing activities;
- supervised - replacement of direct manual control of system operation with computer-directed functions in a supervisory control manner;
- independent - independent, self-actuating, self healing operations requiring a minimal of human intervention.

Telerobotic Servicing System (TRS) Description

At present, the level of robotics and teleoperator technology would probably support applications envisioned by NASA as far as physical performance of necessary functions is concerned. Insofar as a telerobotic approach is required to achieve mission objectives, however, great demands would probably be placed on the system operator in terms of skills and workload. Generally, a robot is distinguished from a telerobotic system in that the former can perform tasks under autonomous local control while the latter requires some level of direct real-time control by a human operator. Remote control by an operator in turn generates requirements for a large amount of information to be fed back from the device and displayed to the operator. The necessary manual controls and the correspondence between controls and manipulator effectors can generate demanding operator tasks in terms of skill levels and of task time and workload factors.

An index of the efficiency of a telerobotic system including all necessary subsystems and control laws is the ratio of time to perform a given task under operator control to the time taken by a human subject to perform the same task by hand. This metric was originally proposed by Vertut, Papot and Rossignol (1973). These authors reported teleoperator to manual time ratios ranging from 1.5 to 100 depending on teleoperator system parameters and motion direction within the reach space for a task involving precise end effector placement and orientation. The system used by Vertut et. al. provided direct vision of the manipulator and task apparatus by the operator. Kirkpatrick, Shields, Malone and Brye (1977) reported time ratios ranging from 21 to 33 for an extendable stiff arm manipulator and from 63 to 127 for an anthropomorphic manipulator with resolved rate control. Both systems provided orthogonal video views of the manipulator and task board. Depending on particulars of the manipulator configuration, the visual feedback system and

the control scheme, therefore, available data suggest that teleoperation may be from 1.5 to over 100 times slower than manual performance of the same task.

Another comparison index for teleoperation versus manual performance of a positioning task involves Fitts' Law (Fitts and Posner, 1967). This law posits a linear relationship between mean movement time and a measure called the index of difficulty which involves the distance and terminal accuracy of a movement. The reciprocal of the slope in the Fitts' law equation can be interpreted as the number of bits of information processed per unit time. When highly practiced subjects perform positioning tasks, a typical result is an information processing rate on the order of 12 bits per second (Fitts and Posner, 1967). Corresponding calculations performed by Kirkpatrick et. al. (1977) using the teleoperator system performance data described previously yielded information processing rates which ranged from 0.07 to 0.74 bits per second. Clearly, even with considerable practice, the act of directly controlling a multiple degree-of-freedom manipulator in real time is inherently difficult.

On the other hand, very effective industrial robots are routinely used for a variety of manufacturing tasks. These cases, however, involve many repetitions of precisely specified movements. When robot technology is applied to manufacturing tasks such as spot welding, for example, the necessary end effector moves and corresponding joint moves are identified in precise detail and the robot is programmed specifically to execute these in a suitable sequence and generally in an open-loop fashion. Such a robot can operate autonomously under an automatic control law and can be fast. It has zero adaptability, however, and must be re-programmed if the task is changed.

These considerations suggest a continuum of human versus automated local control of a telerobotic system. At one end of the continuum would be a pure robot, pre-programmed to execute a few precisely defined movement sequences under automatic control. At the other end would be a pure teleoperator with direct and continuous human control of the manipulator joint angles. The latter system would be adaptable to new tasks and conditions because the human operator exercises a generic level of control and can modify his/her command sequences to adapt to new task requirements or changed task conditions.

The state-of-the-art in telerobotic systems and equipment for spacecraft servicing was identified in the NASA JSC Servicing Equipment Catalog (JSC-22976, 1988). The specific items in the catalog having implications for telerobotic servicing of the Space Station are listed below:

The Flight Support System/Servicing Aid Tool (FSS/SAT) is a remotely-operated bilateral force-reflecting manipulator system. It will enhance the capability of the NSTS Mission Specialists to perform IVA and EVA Free-Flyer Spacecraft servicing in the Space Shuttle Cargo Bay from the Aft Flight Deck. The system, which mounts to the FSS by a versatile electro-mechanical interface, can be repositioned by use of the remote manipulator system. The FSS/SAT will have provision to pickup and restow tools from a tool storage locker mounted on the FSS. The FSS/SAT will either be stowed in a storage rack or will be secured to the FSS during launch and landing operations. The system is designed to operate from the Space Shuttle onboard utilities.

The Light-Weight Module Service Tool (LWMST) is a device to permit remote on-orbit exchange of On-orbit Replaceable Units (ORUs) when coupled to an automated servicer system. It is been redesigned for use with the Orbit Maneuvering Vehicle, Flight Support System/Servicing Aid Tool, Remote Manipulator System, other manipulator and robotic servicery. This tool will permit on-orbit exchange of spacecraft module, payloads, and instrument orbital replacement units. Remote computer or manipulator control is retained to permit the servicing operations to be performed from the Shuttle Aft Flight Deck.

The Orbital Maneuvering Vehicle (OMV) provides for the extension of payload services and capabilities out of the Orbiter and the Space Station. These services include spacecraft delivery and retrieval to and from higher orbits, reboost or deboost, payload viewing and satellite support. The OMV will also be capable of supporting advanced mission kits for remote servicing, refueling, and debris retrieval. On-orbit operations may be controlled either from the ground or the Space Station. Space Stations operations typically will be controlled by the Space Station operator when

the OMV is operating in close proximity to the station. In either case, final docking maneuvers are performed in remotely piloted modes.

The Orbital Spacecraft Consumables Resupply System (OSCRS) is designed to be flexible in order to service a wide range of satellites and be adaptable to support Space Station. OSCRS will also provide adequate data and control to permit independent crew operation/trouble shooting/work-around without ground coverage.

The Payload Berthing System (PBS) provides on-orbit docking/berthing of payloads for servicing, repair or temporary holding. The PBS is sidewall mounted at the primary attachment locations of the cargo bay.

The Payload Interface Mechanism (PIM) mounts on top of the Manipulator Food Restraint (MFR) stanchion. It is a tethering device for attaching a payload to the MFR and consists of three main parts: a payload fitting, a pyramid fitting, and a pyramid housing. The pyramid fitting and the pyramid housing are connected by a retractable tether. Tether attachment rings are provided on the ends of the pyramid fitting's handles and on the payload fitting.

The Remote Manipulator System (RMS) is a mechanical arm which augments the Shuttle systems in performing the deployment and/or retrieval of a payload. In addition, the RMS may be used to perform other tasks in support of satellite servicing or to assist in extravehicular activities. The manipulator arm consists of four joints connected by structural members to a payload-capturing device called an end effector. The movement of the arm is controlled by an operator using a display and control panel and two three-degree-of-freedom hand controllers. The operator also has visual access through the windows in the Aft Flight Deck. The manipulator arm is anthropomorphic by design, comprising shoulder pitch, shoulder yaw, and elbow pitch joints (mainly providing end-point translation) plus wrist pitch, yaw, and roll joints (providing rotation of the end effector).

The Remote Manipulator System Module Servicing Tool (RMS MST) is a device to permit remote on-orbit exchange of On-orbit Replaceable Units (ORUs) when it is coupled to the RMS. It develops high torques up to 160 ft-lbs, with provision for torque takeout and transporting of Multimission Modular Spacecraft (MMS) and other compatible ORUs. The tool is controlled and operated from the Space Shuttle Aft Flight Deck. The RMS MST is modeled after an EVA astronaut operable tool used for the same purpose.

The RMS-Based Handling and Positioning Aid (RMS/HPA) is a mechanical arm which provides a wide range of adjustable work stations both inboard and outboard of the Orbiter Cargo Bay. It is derived from Remote Manipulator System (RMS) technology.

Servo-Actuated Manipulator System With Intelligence Networks (SAMSIN) is a bilateral force reflecting master-slave servo manipulator. SAMSIN is a general purpose electrical-mechanical device. SAMSIN is used to extend the hand and arm manipulative capacity into a "remote hostile" environment. A master-slave manipulator is an extension of the human hand. The remote hand may be used as a tool, but can be used more effectively as the hand that holds and guides a tool. SAMSIN has seven degrees-of-freedom and is bilateral and force reflecting in all degrees-of-freedom.

The Stabilized Payload Deployment System (SPDS) is a dual redundant motorized system designed to deploy RMS type payloads up to 50,000 pounds that are typically secured in the bay with "Port" side and starboard Payload Retention Latch Assemblies (PRLA's) and Active Keel Assemblies (AKA's).

The Standard End Effector (SEE) is the terminal device on the Remote Manipulator System (RMS) arm, and its prime function is to capture, hold, and release payloads. The SEE is a hollow, light-gauge aluminum cylinder which contains a remotely controlled motor drive assembly and three wire snares. The SEE drive system provides the abilities both to capture and release and to rigidize a payload. The SEE is controlled from the RMS control panel in the Aft Flight Deck of the Orbiter.

The Universal Servicing Tool (UST) is a flight power tool that allows changeout of the tool attachments on orbit. Designed to anchor itself to a payload, spacecraft module, or orbiter, the UST can be used to remove or tighten bolts, and operate latches and fasteners while reacting the resulting torque to the anchor points. The UST comprises a control module, a drive module, and interchangeable tool elements. The UST can be operated manually by an astronaut (as a NASA Goddard Space Flight Center version was used on the Solar Maximum Mission SMM), or operated remotely when attached to a manipulator arm.

Telerobotic systems capable of performing the necessary on-orbit servicing functions will probably require at least four general capabilities as follows:

- mobility providing transit aboard or around the Space Station. This capability will be provided by a free flying maneuvering vehicle such as the OMV, or by dexterous manipulator systems mounted to rails attached to the Space Station, or by being capable of being moved and installed at specific attachment points on the exterior of the Space Station or Space Station elements
- manipulation involving multiple degree-of-freedom manipulator arms and end effectors suited to the required servicing activities
- sensing and communications to provide sensor feedback to the remote system operator concerning the task in progress, navigation data, environmental data, and manipulator/end effector status, position and orientation data
- control and computational which directs motors or other effectors in the mobility and manipulator subsystems according to automated control schemes, operator commands, or some combination of the two.

Space Station Flight Telerobotic System (FTS)

The FTS is designed to be a teleoperated device controlled by a crew member from within the Space Station itself or the National Space Transportation System. Limited autonomous capability is projected. The two principle components are the telerobot and workstations. The FTS will require transportation to and from worksites. This will be available from the Mobile Transporter, the Mobile Servicing Center, and EVA crew members. The FTS will rigidly stabilize to the worksite, and all structural loads will be transferred to the worksite. Power, data, and video accomodations will be available either at the worksite or from the FTS. The FTS will be useful for missions that are outside the safety envelope for human operators. Scenarios requiring excessive strength, reach, and duration may be better accomplished with the FTS. The FTS will be capable of detecting failures and automatically safing. In addition, an EVA crew member will be able to shut down the telerobot with a redundant direct link. The telerobot is designed to be operable by one person at the workstation; bilateral force reflection between the telerobot and the hand controller shall be provided. Man-machine interfaces necessary for control of the telerobot will be designed into the system.

An example mission for FTS Space Station servicing was described by Malone and Permenter (1987 and 1989) who assessed the requirements for FTS servicing of the Gamma Ray Observatory (GRO). The GRO refueling mission involves capturing the satellite with the Remote Manipulator System (RMS), placing it on the Flight Support System (FSS), mating the refueling hoses, pumping the fuel, demating the hoses, grappling GRO with the RMS again, and returning her to orbit.

The fuel used to power the Gamma Ray Observatory is harmful to humans. The potential for coming in contact with the fuel during the refueling scenario has to be reduced. The refueling mission is a time consuming effort due to the fuel used to propel the satellite. Transferring the fuel too rapidly will cause it to heat up beyond acceptable temperatures.

Capturing GRO and placing her upon the FSS may require an extensive amount of time, producing the possibility that the refueling scenario will occur over a couple of days. The first day will be devoted to capturing the satellite, rotating it 180° and placing it on FSS with the help of a camera on the base of GRO. Refueling will then be accomplished on the second day of the mission.

Mating the couplings with FTS involves only one operator who remains inside the pressurized cabin. The movements of the robotic machinery are slow and awkward, yet more efficient than EVA operations which require excessive front-

end preparations. The precision required when mating the couplings is difficult to achieve telerobotically, however the danger of exposure to the propellant makes FTS a likely candidate for the procedure. Refueling GRO by FTS operations takes 29.4 hours and involves one crew member. This crew member is working one-hundred percent of the time, and has completed one-hundred tasks by the time the satellite has been refueled.

Human Factors Issues in Telerobotic Servicing

The significant issues in human factors design of telerobotic systems for Space Station servicing are: definition of the role of the human in the control of the servicing operation; operator-telerobotic device interface design; and development of a simulation test bed to develop and validate operator-machine interface design approaches.

Definition of the Role of the Human

The issue in the determination of the role of the human in telerobotic systems control involves the development of alternate techniques for allocating system functions and subfunctions to human and machine performance. Specific allocation techniques will include:

- Allocations to man or machine
- Allocations of human operator functions among several operators
- Allocations to local or remote control
- Allocations to control modes ranging from automated to manual teleoperated control.

Feasible allocation approaches will be determined through an assessment of the expected effectiveness of each candidate approach in terms of system requirements. Where optimal allocation decisions can be made based on existing data, these decisions will lead to allocation concepts. Where additional data are required to complete the allocation, or to verify an allocation decision, a requirement for simulation data will be generated. Control modes include: 1) automated control, typical of industrial robots where the system performs in a completely pre-programmed manner; 2) adaptive control, wherein the telerobotic system learns to adapt to its sensed environment with tutoring from the human operator, thereby developing its own rules and algorithms; 3) supervisory control, wherein the system performs pre-programmed or adaptive routines and response under supervision of the human operator or executive software; 4) interactive control where the operator and the computer cooperatively share control authority under specific interaction protocols; and 5) manual teleoperated control.

The allocation of subsystem control functions to operator or machine should take advantage of capabilities including local autonomous action and supervisory control so as to lighten the burden on the human operator relative to the current level of activity and skill required in pure teleoperator systems.

Supervisory control implies one of or both of the following:

- A manipulator system performs the bulk of the movements required for a given task under local autonomous control laws. This performance is monitored by a human operator who assumes direct control only if required.
- A manipulator system has a series of subroutines which produce standardized "elemental moves. " The operator commands the device during task performance by selecting and stringing together elemental moves.

Application of either or both of these closely related versions of supervisory control has considerable potential for reducing the burden on the operator of a telerobotic system and for increasing the speed and accuracy of task performance. Application of supervisory control techniques is expected to facilitate development of the remote telerobotics capabilities envisioned by NASA. The human factors issues inherent in the implementation of supervisory control for telerobotic systems were addressed by Malone, Kirkpatrick and Seamster (1988).

Identification of Operator-Device Interface Design Requirements

Operator-machine interface requirements will be developed for displays, controls, consoles, workspace, telepresence, communications, user-computer interfaces, and procedures.

- Displays
  - Situation displays (resource management, sighting devices, and detection aids)
  - Environment displays

    - video (stereo vs. mono, FOV envelopes vs. detail resolution requirements, pointing, integration, point of view)
    - special sensors (tactile, kinesthetic, motion detection, and proximity - obstacle detection)
  - Navigation displays
    - displays of arrangements (route planning, destination planning, geographic reference)
    - location, position and orientation of each manipulator
    - formation displays
    - standard routines (transit, evasion, positioning and orientation, synergistic formation, and stored movement sequences)
  - Status monitoring (temporal, diagnostics, and voice display)
  - Data base access displays
  - Communications displays
  - Expert system interface
- Controls (manipulator, mobility systems, sensor control, testing/troubleshooting, communications, navigation systems, mission data processing, and zero-g and operating envelope constraints)
- Consoles/panels (control and display arrangements, panel packaging, and zero-g and workspace volume constraints)
- Workspace (workstation, arrangements, seats/restraints, access/egress, windows, and safety devices)
- Telepresence (display integration, whole or part task, visual only or multisensory, panoramic or multiple camera viewing, and multisensor integration)
- Communications (mode control, status monitoring, link optimization, and uplink/downlink characteristics)
- User-computer interface (control authority, user-computer dialogs, continuous control, protocols, expert system interfaces, special displays, decision aids, and training modes)
- Procedures (sequences, job design, job aids, and decision rules)

Development of an Operator-Machine Interface Simulation Capability

While a considerable body of theory and research exists concerning autonomous, semi-autonomous and supervisory control, application of these techniques to the operator-machine interface of real systems is not routine and is not supported by a corresponding body of proven engineering and human factors practice. Therefore, it is viewed as essential to the applications envisioned that a man-in-the-loop simulation capability be developed to support evaluation of feasibility and effectiveness of advanced concepts for allocation of control functions to operators and machines.

Specific simulation requirements include such issues as degree of simulation fidelity, simulation data requirements, data acquisition and recording requirements, and data analysis requirements. The specific issues are: 1)Degree of simulation (whole task vs. part task, extent to which controls and displays are real, simulated or dummied, engineering simulation vs. prototyping simulation vs. procedures, and development simulation vs. training simulation); 2) Determination of simulation data requirements including type of data (Performance measures, control variables, and independent variables), data reliability requirements (experimental controls and independent variables), data validity requirements including sampling criteria for missions and conditions, operations and tasks, operators, and design concepts, and fidelity requirements; 3) Data acquisition and recording requirements including data acquisition instrumentation, data integration, and monitoring vs. measurement data; 4) Data analysis requirements including quick look analysis, performance analysis and trends, inferential statistics and descriptive statistics; 5) Simulation visual subsystem including view of manipulators from the perspective of the control station, view of the individual manipulators from the perspective of another manipulator, view of the target system from the perspective of a manipulator, and view of the Space Station from the perspective of a manipulator; and 6) Manipulator control simulation enabling simulated control of manipulator arms and end effectors with an integrated view of the worksite.

References

Fitts, P.M. and Posner M.I. Human Performance. Brooks/Cole, New York, 1967.

Kirkpatrick, M., Shields, N.L., Malone, T.B. and Brye, R.G. Manipulator System Performance Measures. Mechanism and Machine Theory, 1977, Vol.12. pp 439-450.

Malone, T.B., Kirkpatrick, M., and Seamster, T.L. Operator Interface Issues in theControl of Telerobotic Systems, Society of Manufactoring Engineers Robotics 12 and Vision 88 Conference, June, 1988.

Malone, T.B. and Permenter, K.E. Tradeoff of EVA and FTS for Space Station Servicing,National Aerospace Electronics Conference (NAECON) , 1988.

Malone, T.B. and Permenter, K.E. Development of a Decision Aid to Support the Selection of EVA or FTS in Space Station Servicing, NASA JSC, 1987.

McDonnell Douglas Corporation The Human Role in Space (THURIS) Marshall Space Flight Center, 1984. NASA JSC Servicing Equipment Catalog (JSC-22976, 1988).

Vertut, J., Papot, L. and Rossignol, C. Contributions to Define a Dexterity Factor of Manipulators. Proc. 21st Conference on Remote Sensing Technology, 1973.

INTERACTIONS BETWEEN THE SPACE STATION AND THE ENVIRONMENT

A Preliminary Assessment of EMI

G. B. Murphy
H. B. Garrett

Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, California 91109

ABSTRACT

A review of the interactions between proposed Space Station systems/payloads and the environment that contribute to electromagnetic interference has been performed. Seven prime sources of interference have been identified. These are: The Space Station power system; active experiments such as beam injection; ASTROMAG; ram and wake density gradients; pick up ions produced by vented or offgassed clouds; waves produced by current loops that include the plasma and structure; arcing from high voltage solar arrays (or possible ESD in polar orbit). This review indicates that: minimizing leakage current from the 20 kHz power system to the structure; keeping the surfaces of the Space Station structure, arrays, and radiators non-conducting; minimizing venting of payloads or systems to non-operational periods; careful placement of payloads sensitive to magnetic field perturbations or wake noise; and designing an operational timeline compatible with experiment requirement are the most effective means of minimizing the effects of this interference. High degrees of uncertainty exist in the estimates of magnitudes of gas emission induced EMI, radiation of 20 kHz and harmonics, ASTROMAG induced interference, and arc threshold/ frequency of the solar array. These processes demand further attention so that mitigation efforts are properly calibrated.

1.0 INTRODUCTION

The Space Station, as a resource laboratory for a wide range of scientific experimentation, must provide an environment compatible with many (sometimes conflicting) objectives. The purpose of this paper is to summarize an investigation into the major sources of contamination of the external electromagnetic environment. This environment, specified in SSP 30420, limits narrowband and broadband electric fields to levels illustrated in figures 1a and 1b and limits magnetic fields to levels shown in figure 2. The ElectroMagnetic Environment (EME) requirements go beyond that traditionally accepted for space-borne equipment (MIL-STD-461C, part 3). The reasons for this difference are not particularly mysterious; the requirements for the Space Station are driven simply by a need for low background emission for sensitive experiments instead of the receiver interference and electronic compatibility issues encountered in typical military or space hardware.

The Space Station is of unprecedented size, carries experiments that can disturb or interact with the background environment, has a power system much different than has ever been flown on a spacecraft, and outgasses/vents products which affect the environment. All of these factors must be considered in assessing which particular design options, hardware configurations, or operational scenarios may adversely affect the EME, and cause the station to be an unsuitable carrier for certain instrumentation. We have examined possible interactions between the hardware or effluents and the natural environment. From this examination, we have identified seven processes that may adversely alter the EME. These processes are:

a. leakage of 20 kHz and harmonics from the power system,
b. waves induced by ionospheric currents closing through the large Space Station structure,
c. possible plasma trapping and EMI generated by the ASTROMAG superconducting magnet
d. broadband electromagnetic noise from possible arcing of the solar array,
e. Ram/wake plasma density gradients,
f. ionization of effluent clouds,
g. waves induced by particle accelerators (e.g. electron beams)

We shall examine each of these processes in detail, assess the impact of each on EME, and recommend courses of action that minimize the effects.

The specification for Space Station defines field disturbances according to broadband electric, narrowband electric, and narrowband magnetic. Before we begin detailed discussion of each of the physical processes, we shall define more precisely the meaning of these terms. This will allow us to determine which processes impact which specifications.

'Broadband emissions' can be created by two fundamentally different processes. The first is what we shall call impulse noise. That noise is generated by a system producing a pulse of current in a conductor that is short in the time domain (broad in the frequency domain). This noise typically has its highest frequency component inversely proportional to the rise time, and its lowest frequency component inversely proportional to the duration of the pulse. The voltage phase is coherent across the band. This type of noise can be produced by relay closures, arcs, etc. When detected by an 'antenna', the noise voltage will be proportional to the bandwidth (BW) and the noise power proportional to $(BW)^2$.

A second type of broadband emission is continuous in both time and frequency. When observed on an oscilloscope or listened to in the audio frequency domain, it appears as 'white' noise. This emission results from physical processes associated with the thermal motion of electrons in electronic systems, and, as we shall see, from certain plasma processes. The spectrum is not always 'white'; that is, the frequency domain may reveal slopes or cutoffs but the noise is broadband in the sense that the noise frequency components are continuous across the band of interest. This type of noise has the detected voltage proportional to $(BW)^{1/2}$ and detected power proportional to BW and differs from impulse noise in that its phase is random across the band. Broadband noise of this nature is particularly annoying in communication systems because the signal to noise ratio, assuming the desired signal is narrowband, is inversely proportional to BW.

Narrowband emissions, both electric and magnetic, can be regarded as continuous in the time domain and sharply peaked in the frequency domain. Detected voltages add in an RMS manner since different sources are (in general) incoherent. Likewise the signal power, once within the receiver bandwidth, is independent of bandwidth.

It should be noted that the wide variety of signals encountered in nature are not always so easily classified. Many processes produce noise with power spectral density proportional neither to BW or BW^2 but somewhere in between. In examining interactions between Space Station components and the natural environment, it will be necessary to classify sources of interference that fit none of these definitions precisely.

We must further note that although the process of modulating current through a conductor produces both electric and magnetic fields, not all of the processes producing waves in a plasma produce electromagnetic waves. Many produce only 'electrostatic' waves. These are not like free-space electromagnetic waves and, along with having no magnetic component, may have their electric field along the direction of propagation (or at some angle) instead of transverse. An excellent review of plasma waves occurring in nature is given by Shawhan [1985]. Many plasma waves do not propagate at all in the strictest sense but their electric fields must be considered in interference calculations just as one must consider near-field evanescent waves near a dipole antenna. We shall, in our analysis, consider interference fields near the Space Station structure and not concern ourselves with propagation effects unless appropriate.

2.0 RADIATION OF 20 KHZ AND HARMONICS

This radiation, which will be both electric and magnetic, is from two sources. The first source is leakage because of imperfect shielding of the transmission line. The second source of fields are those produced by current that is present in the structure (chassis) and exists because of finite impedance between elements of the power system and structure ground. As we shall see, this chassis current can easily dominate the EME at 20 kHz and harmonics.

2.1 RF Radiation

Let us first consider 20 kHz radiation from the transmission line. A number of studies have focused on the trade-offs required in choosing a power system for Space Station [Hansen, 1987; Rice, 1986; Simon and Nored, 1987; Renz et al, 1983], but few have dealt in more than a qualitative way with the potential for EMI. One study by Pistole [1985] was focused on EMI considerations but used three-phase 200 volt AC for the primary and assumed flat bus bars for distribution in the modules. As with all EMI analyses, the end result is very sensitive to system configuration. The transmission line being considered for 20 kHz primary distribution is a double-sided strip line design [Schmitz and Biess, 1989]. Here we examine the work of Schmitz and Biess [1989], since those measurements reflect most accurately the current system configuration.

The Schmitz and Biess tests were performed in a screen room with the cable driven by a

prototype 20 kHz resonant inverter. Electric fields were measured with the source end at 440 V and the load end open. Magnetic fields were measured with a resistive load designed to draw 60 amps. The sensor loop was placed in three planes: one parallel and two perpendicular to the transmission line. Figure 3a shows the average radiated magnetic field (at one meter) along the length of the transmission line and compares it to the current SSP 30237 limit and source current. Figure 3b illustrates the measured narrowband electric field (also at one meter) and compares it to 30237. The harmonic content of the current and voltage may be load dependent, therefore these tests must be used for comparing harmonic content of the emissions to that of the source. Note that, indeed, the radiated field is closely related to the source current and that high frequency emission is somewhat enhanced. Unfortunately, the power system design is not yet firm enough to state that these measurements represent what can be expected in the completed system and only give us a first order estimate.

2.2 Chassis Current

AC currents may also be induced in the chassis by stray capacitance between the cable, converters, etc. and the chassis. These currents must be distinguished from those in the transmission line. Stray currents that traverse the length of the Space Station structure cause the structure to behave like a loop antenna. Whether the structure is insulating on its surface or conducting determines the interaction between this 'antenna' and the plasma.

If the structure is conductive it will have a significant sheath surrounding it due to the v x B motional potential. This sheath has been shown to be capable of conducting noise over large distances very efficiently [Laurin et al, 1989]. Sheath waves are guided waves that are conducted along conductors surrounded by sheaths much like waves in a coaxial cable transmission line. Anywhere sheaths overlap, the waves can propagate. The significance of this is that noise generated locally can be conducted along Space Station structures to other cables which may be sensitive to this frequency. Unless the sheath is forced to collapse, the waves propagate with little attenuation. Therefore, as a worst case scenario, we assume that cables placed anywhere externally on the Space Station may be within a sheath which is "connected" to a source of noise via the "structure-sheath coax transmission line". The electric field and magnetic field within the sheath depend on the size of the sheath, various plasma parameters, and the frequency of interest. Cut-off for this propagation, when the structure and sheath are in a magnetic field, will be somewhere near 1/2 f_e (electron gyrofrequency) or about 500 kHz. This allows propagation of 20 kHz and the principal harmonics. The radial and longitudinal components of the electric field change as the frequency increases but, for 20 kHz and the third harmonic, it will be mostly radial.

It is very difficult to predict the magnitude of the interference. A worst case estimate calculates the electric and magnetic fields near a current loop where the value of the current is chosen to be the expected leakage current. The distance from the loop is chosen to be the sheath size (~10 cm) instead of the actual physical separation. In the case of Space Station, the loop is chosen to include an electrical element, such as the cable or cable tray, and the truss structure. Assuming a worst case loop area of 100 m^2 (2m x 50m) and a measurement distance of 10 cm, the E and H fields in the sheath may be approximated by:

$E \simeq 40 + 20 \log (I_1/I_0)$ dBv/m at 20 kHz

$E \simeq 50 + 20 \log (I_2/I_0)$ dBv/m at 60 kHz

$H \simeq 200 + 20 \log (I/I_0)$ dBpT

where I_0 is 1 amp and I is the assumed leakage current

Thus for an allowed leakage current of 1 ma the worst case fields observed in the sheath would be:

$E \simeq -20$ dBv/m

$H \simeq 140$ dBpT

It is suggested that a serious effort be undertaken to determine the affect of geometry, to analyze the effects of insulating the struts to minimize sheaths, and to develop methods for ground test, so that the extent of this problem of narrowband electric/magnetic field interference may be determined and appropriate susceptibility tests be developed. If it is possible to ensure attenuation of sheath waves, much higher leakage currents can be allowed. For example, the field at 1 meter from the ground loop discussed above is down by 40 dB for electric and 60 dB for the magnetic components which, although still an issue when compared to spec, are much more tolerable from an interference standpoint. Making the surface of the struts non-conducting will reduce their sheath and help this problem.

3.0 IONOSPHERIC CURRENT CLOSURE

In addition to AC currents coupled to the structure by the power system, parts of the structure which are uninsulated conductors

can couple to the ionosphere causing DC current flow.

The DC current flow is induced by the potential difference (with respect to the plasma rest frame) between different ends of the conductor. This potential is of magnitude

$$\phi = v \times B \cdot l$$

where v = spacecraft velocity vector
B = magnetic field vector
l = vector distance between points that contact the plasma

If the conductor is exposed along its length, the electric field in the sheath around the conductor can induce lower hybrid waves [Hastings et al, 1988]. The part of the conductor that collects current from the plasma causes the production of Alfven waves.

It is not our purpose in this paper to discuss the physics of how conductive objects moving through a magnetic field in the presence of a plasma produce waves. The reader is referred to Barnett and Olbert, [1986]; Hastings et al, [1988]; for a discussion of the production of lower hybrid waves by AC currents in the structure. Drell et al [1965] is a good source for an introduction to the phenomena of Alfven waves induced by passive current collection. Acuna and Ness [1976] observed these waves in the Jovian environment. Our brief discussion here is based on these and other references in the context of the Space Station.

3.1 Alfven Waves

The Alfven wave is a hydromagnetic wave stationary in the Space Station reference frame. The power loss due to this wave (and thus its magnitude) depends on the conductive area perpendicular to B and factors that determine current collection such as surface potential and plasma density. An analysis of the passive DC currents induced by motional EMF in the Space Station system, assuming the solar array surfaces and modules are conductive and the structure is non-conductive, was performed. This is a worst case scenario and the results can be summarized as follows.

Power loss (drag) for Space Station is limited by ion current collection in the ram direction and photoelectron emission in the wake for the altitude range of 200-400 km. If the Space Station solar arrays are conductively coated and bonded to the chassis, the current limit is about 500 ma (eight wings at 60 ma each). This results in a power loss of .3 watts for a plasma density of $2e5/cm^3$ and in an electromagnetic drag which is small compared to the aerodynamic drag.

The magnetic field in the Alfven wing will have an average magnitude (at $2e5/cm^3$) of about 5 nT indicating that sensitive magnetometers which typically desire noise levels of .1 nT may be affected and must be carefully placed to avoid the wings.

Although when doppler shifted into the Space Station reference frame, the Alfven wave becomes stationary, the plasma density, current collection area, and magnetic field spatial and temporal variations cause the Alfven wave to have low frequency components. An upper bound for these variations is a DC value of 5 nT. Spatial variations will have a frequency cut-off for values higher than v/L where L is the characteristic array (or current collector) dimension and v is the spacecraft velocity. This is between 50 and 100 Hz for Space Station.

Thus we see that the worst case Alfven wave disturbance creates DC and low frequency components of the magnetic field. This disturbance will most likely be an issue only for sensitive magnetometers that attempt to measure currents in the plasma or map the finely detailed temporal variation of the earth's field. These worst case fields are produced assuming that the solar arrays, placed at each end of the structure, are conductive and tied to chassis. This allows for a large v x B·l potential and maximum current collection. If the arrays are conductive but not tied to chassis these worst case fields can be reduced by about a factor of 5. If the arrays and structure are insulated from the plasma and the Space Station is grounded to the plasma at a central location (e.g. the pressurized modules), the fields are reduced almost an order of magnitude. Careful placement of magnetometers may avoid the disturbed field in the Alfven wing, but a detailed analysis will need to be performed once the Space Station geometry and conductivity of its many surfaces are determined.

3.2 Lower Hybrid Waves

Lower hybrid waves are electrostatic waves with their electric field approximately perpendicular to the local magnetic field. They can be excited by the components of sheath electric fields perpendicular to B which exist around conductive Space Station structures. Both DC and AC components of the sheaths can excite such waves. Hastings and Wang [1989] analyze this process in detail for the Space Station case and note that the radiation generated (in the far field) depends sensitively on the geometry and conductivity of the

structure. Barnett and Olbert, [1986] and Hastings et al [1988] also discuss this wave generation mechanism.

The component of radiation due to the DC sheath (and a DC current flow through the structure) is a continuum emission. That is, it is pseudo-broadband and, in the plasma rest frame, will exist in the frequency range $f_{eh} < f < f_e$ or, for the Space Station

$$5 \times 10^3 \text{ Hz} \leq f \leq 9 \times 10^5 \text{ Hz}$$

Although the references cited above analyze the radiation produced in the far field rest frame and we are interested in the near field moving frame, some of the results can guide us in designing a system which minimizes the generation of these waves.

The power loss (Hastings and Wang [1989]) can be written as:

$$P_{rad} = I^2 Z = \left(\frac{vxB \cdot l}{Z_{rad}+R} \right)^2 Z_{rad}$$

where Z_{rad} = radiation impedance
R = impedance of structure
$vxB \cdot l$ = motional potential
I = ionospheric current closing through structure

This can be minimized by decreasing the current collected from the plasma (decreasing collecting area, decreasing motional potential) and by maximizing the mismatch between the structure impedance and Z_{rad}. Power loss has been calculated by Hastings [1989] to be on the order of a watt for reasonable values of structure resistance and a geometry that has solar arrays with conductive surfaces. This is similar in magnitude to worst case power losses calculated for the Alfven waves.

Since Z_{rad} is very sensitive to geometry and plasma composition, the best approach for minimizing this noise source seems to be to limit, as much as possible, the current collection which is consistent with the recommendations of the previous sections. Calculation of the Doppler shifted spectrum and an estimate of wave magnitudes has not yet been completed and remains to be addressed theoretically.

4.0 ASTROMAG

The large superconducting magnet, ASTROMAG, accepted as an attached payload on Space Station, has been analyzed to determine the levels of electromagnetic disturbance. The DC magnetic field, possible effects of quenching, plasma wave emissions, and helium leakage have all been examined. The former two will be described in more detail in the following paragraphs. It should be noted that it is virtually impossible to predict with an accuracy better than an order of magnitude what wave emission levels may be. The interaction between the ASTROMAG magnet and the ionosphere constitutes a fundamental plasma experiment which has not been performed in the laboratory. Bounds can be placed on the available energy for wave emission but it is not possible to assess how much of the energy is channelled into any particular wave mode without complex model development.

4.1 DC Fields

The magnet is set up for nominal operation as a quadrupole so as to minimize the resultant torque by the earth's field [Sullivan et al, 1989]. The remaining torque is comparable to aerodynamic drag torque assuming a 30-40 meter distance from Space Station center of gravity. The DC fields will, however, obviously affect sensitive magnetometer measurements. The coil's field reaches a level equivalent to the earth's field at a distance of 15-20 meters. Since this field falls off as r^{-3}, at a distance of ~75 meters (which is about as far away as you can get from ASTROMAG), the field contributes $\sim 2.5 \times 10^{-4}$ G or about .1% to the background. If this interference field were constant, it should be possible to subtract it from any measurement. However, it is important to realize that in order to subtract this interference field one needs to know alignment accurately. For example a 1° alignment error results in a change of several hundred nT at 20 m which is considerably greater than the signals measured by sensitive magnetometers. Additionally, if alignment changes are due to thermal and dynamic effects, there will be a time varying component to this field. If sensitive magnetometers are flown, they should be located as far as possible from ASTROMAG and the magnet may have to be off for their measurements.

4.2 Quench

If the coil should suddenly lose its superconductive properties (e.g., loss of coolant, shorted coil, micrometeroid impact, etc.), the magnet will quench. How a superconducting magnet quenches is part of its design. A probable I(t) during quench has been obtained from the Magnet Lab at MIT for a typical design configuration. The maximum dI/dt is ~1000 amp/s and the characteristic decay time is ~1 s. This quench is quite slow compared to the 10^{10} amps/s dI/dt and nanosecond rise times for ESD events. Radiation from this process would appear to be of low frequency and pose no hazard to Space Station or payload systems. It is very important, however, that this quench be treated

carefully during instrument development to assure that no failure modes are introduced that allow faster current rise times. Rise times 10 - 100 times greater may begin to be of concern. The effect of the quench on the plasma confined in its magnetosphere has not been analyzed.

4.3 EMI from Plasma Processes

As discussed above, the ASTROMAG magnet is itself an interesting plasma experiment. We have studied the various mechanisms that could lead to plasma energization and conclude that it is likely that a substantial plasma density can build on the closed field line region and that a significant fraction of electrons will be accelerated to energies high enough to cause molecular excitation and generation of a broad spectrum of waves. Since we cannot explicitly predict the wave energy likely in a specific frequency band, we have estimated the total energy available for excitation processes. The result, assuming a background ionospheric density of 10^5 electrons/cm^3 and 10^8 neutrals/cm^3, is that the two sources of free energy, impinging neutrals and ions, are estimated to contribute 20 - 200 mwatts of energy to waves and optical emissions near the Space Station.

Table I summarizes the possible types of radiation, the frequency ranges, and potential sources of plasma waves. Only the lower hybrid and cyclotron waves can be bounded in magnitude based on analogous measurements of wave energy induced by pick-up ions on the Shuttle [Gurnett et al, 1988]. This magnitude is ~1 mv/m and has been classified as narrowband even though it occurs over a broad frequency range.

No emissions are expected to be at a level high enough to interfere with electronic systems but they may interfere with sensitive instruments by raising the background noise level. Only two precautions can be taken to minimize EMI (and other effects such as glow) from the magnet. First, minimize gas emissions (especially species with low ionization potential and easily excitable metastable states) near the magnet's "magnetosphere" and second, simply turn the magnet off if it creates background noise that is unacceptable to other experiments. Designers of the magnet as well as the operational timeline should be sensitive to these issues.

5.0 ARCS AS A SOURCE OF BROADBAND NOISE

Arcs are transient events that produce true broadband electromagnetic noise. In the low altitude low inclination orbit of Space Station the only serious candidates for environmentally induced arcs are the photovoltaic arrays. A number of experiments, notably the PIX flight experiments [Grier, 1985; Purvis, 1985; Ferguson, 1986] have studied the problem of arcing for negatively biased solar arrays. Two fundamental questions remain unanswered: 1) How does the arc onset voltage depend on cell geometry, and on the background plasma/neutral density/composition? 2) How does arc rate scale with these parameters?

Only two theories known to the authors address these issues. Jongeward [1985] suggests that a contaminant insulating layer on the interconnects interacts with ions collected from the plasma to produce fields strong enough to generate high electron emission currents leading to avalanche ionization. Hastings et al [1989] theorize that gases desorbed from cover glasses by electron bombardment produce a neutral density in the vicinity of the interconnect that is high enough to lead to breakdown. Unfortunately, results of preliminary experiments conducted on Space Station solar cells are not yet available. 160 V was chosen for the operational voltage primarily because no arcing was observed with the PIX array below 200 V. However, since we do not yet know definitively how the phenomena scales with cell geometry and environmental conditions, we can not be certain that -160 V is below arc threshold. Validation must wait until tests are completed under realistic flight conditions.

Experiments with older cell geometries suggest that the arc onset voltage and frequency may be dependent on plasma and/or neutral density [Snyder, 1984]. Both theories suggest that background neutral density as well as plasma density and composition may be critical. The Hastings et al [1989] theory suggests that temperture may be a factor since it affects outgassing. We begin to see an example of a synergistic effect. Thruster operations, local offgassing, and ram surface pressure all act to enhance the local density, as would any environmentally induced outgassing. The worst case envi- ronment is (even without thruster gas effects) expected to show about one order of magnitude increase in plasma density and about two orders increase in neutral density as a result. For the purpose of this paper, we therefore assume by extrapolation of current data [Grier, 1985; Purvis, 1983; Snyder, 1984] that the array could arc and estimate the magnitude of the interference generated. Leung [1983] has conducted experiments in an acrylic anechoic chamber where the arc spectrum and intensity for a given arc current have been measured. We shall use his results to scale to Space Station after calculating the probable arc magnitude.

Kuninaka et al [1986] have suggested that the emission of electrons from the dis-

charge sites is determined by space charge limited current flow. However, the value for area and distance used in the calculation is uncertain. Experiments have shown [Snyder, 1984] that the peak current seems to be related to the value of the capacitance chosen. Up to 50 amp has been measured by Miller [1985] and there was evidence that interconnects showed damage due to melting of the metal surface. A real array, when powered up, will supply approximately 2 amps (~3 amps for short circuit) before limiting. All experimental evidence suggests that an arc, once initiated, will draw the current necessary to bring the bias below the point where the arc will cut off. The limit is probably based on the details of the emission characteristics at the arc site. We therefore assume that for the Space Station array an arc will bring one sector (16 cells at 8 volts and 2 amps) to a cut-off condition.

We can now use Leung's data on radiated emissions to estimate the Space Station electromagnetic environment. Leung's data on EMI were taken for peak currents estimated to be on the order of .1 to .2 amps. Therefore, we shall scale his data by a factor of 10 for worst case Space Station array arcs. Figure 4 scales the laboratory data to Space Station assuming a measurement distance of 20 meters (Leung's was 1 m). Although the radiated levels are not enough to disturb or damage electronics, they will be ~50-90 dB above the Space Station broadband spec. Note also that this noise is electromagnetic and the impulse nature of the arcs can present shielding difficulties for the magnetic component. Even for the very low probability of an individual cell arcing, the number of cells in the Space Station photovoltaic arrays imply a serious source of interference.

A preliminary assessment of conducted emission on the transmission line due to solar array arcs has been done by Stevens et al [1986] and they find no adverse effects. More detailed analysis has been done by Kuniaka and Kiriki [1989] to determine induced circuit transients. They also conclude that arcs of less than 100 V should produce negligible conducted interference. The analysis needs to be repeated, however, once power system models are more mature and verification tests are complete on the Space Station cells.

6.0 WAKE TURBULENCE

Although numerous papers have addressed the physics of the plasma wake at mesosonic velocities, few have discussed the EMI that can be generated. Leung [private communication] has measured Diachotron waves in the laboratory. Ma et al [1987] have reported electrostatic noise generated in the wake of Titan (Voyager observations). Recently Tribble et al [1989] have reported on plasma turbulence and electrostatic noise in the Shuttle wake. Unfortunately, it is very difficult to scale with certainty either the laboratory or space measurements to Space Station. Although Shuttle is close in scale size to the Space Station and flies in a similar orbital environment, it is surrounded by an offgassed cloud which itself generates plasma turbulence and electrostatic noise (see section 7.0). Therefore using the Shuttle data as an upper bound, we obtain figure 5 for the worst case wake-induced noise. It is important to note that this noise is confined to the region near the ion mach cone. Objects on the truss that are tens of meters away from the solar arrays, or other large objects such as the pressurized modules, should see noise of considerably less magnitude and be affected only by smaller wakes of objects more local. Models of wake noise generation and propagation are too immature to refine the estimate further.

7.0 GAS CLOUD EMISSION

Recent Spacelab experiments aboard the Shuttle Orbiter have provided a wealth of heretofore unobtainable information about the interactions between large bodies and the LEO plasma. The Shuttle is not only the largest body flown to date but, as was discovered over a period of time, carries with it a large gas cloud. The discovery of "Shuttle glow" [Banks et al, 1983], broadband electrostatic noise [Shawhan et al, 1984a], heated electron populations [McMahan et al, 1983], a modified ion environment [Hunton and Carlo, 1985], and contaminant ions in the wake [Grebowsky et al, 1987] have begun to fill in pieces in what appears to be a complex puzzle associated with large body induced environments and contaminant interactions. Recent studies of the neutral and ion populations during thruster operations [Wulf and Von Zahn, 1986; Narcisi, 1983; Shawhan et al, 1984b], modification of the plasma during FES operations and H_2O dumps [Pickett et al, 1985; Pickett et al, 1988], the discovery of pick-up ions consistent with chemistry of the H_2O, O^+ interaction [Paterson and Frank, 1989] as well as observations by neutral mass spectrometers [Hunton and Swider, 1988; Wulf and Von Zahn, 1986; Miller, 1983], have helped to sort out the interactions which result from release of contaminants by the Orbiter. Observations by IR, optical, and UV instruments on board the Orbiter [Torr, 1983; Torr and Torr, 1985; Torr et al, 1988; Koch et al, 1987] and by IR on the ground [Witteborn et al, 1987] have provided insight into the effects of both absorption and emission by this contaminant population. Ground observations of shuttle

plumes and modeling of their interaction with the background plasma by Bernhardt et al [1988a; 1988b] have given additional insight into the ionization of contaminent clouds. It is now clear, as a result of these pathfinder experiments, that to conduct experiments in plasma physics, provide long-term monitoring and a data base for the ionosphere, observe astronomical targets over a broad range of wavelengths, and provide sensitive remote sensing capability, the Space Station environment must be kept free of neutral gas emission.

The EMI which can result from these gas clouds is related to their ionization by charge exchange, collisions, solar UV, or CIV processes and the currents these ions produce.

Murphy [1988] has examined published data from the Plasma Diagnostic Package on the OSS-1 and Spacelab 2 missions and correlated the level of pseudo-broadband electrostatic noise with emission of water vapor. The water, which easily charge exchanges with the background 0^+ plasma, produces a ring distribution unstable to the growth of electrostatic waves [Hwang et al, 1987; Pickett et al, 1985; Gurnett et al, 1988].

The level of noise at 1 kHz (chosen as typical of the pseudo-broadband noise spectra for these data) is plotted in figure 6 for three different cases of "small" gas cloud releases. The level of uncertainty in the measurement of H_20 density is represented by the vertical error bars. The three cases chosen represent almost 3 orders of magnitude in gas density. In all cases, the dominant gas is H_20. The first is the H_20 vapor cloud associated with the Orbiter outgassing per se, the second, an operation of the Flash Evaporator System (FES), and the third, a typical operation of a VRCS thruster. In all cases the releases were on the dayside and in an ambient 0^+ plasma of density $\sim 10^5$ cm^{-3}. Note that the data indicate that the noise is linearly proportional to the density of gas released. The best fit to the data is that the intensity (at 1 kHz) of electrostatic noise is proportional to the product of H_20 and 0^+ densities. The constant of proportionality is such that at a 1 g s^{-1} release rate, the measured electric field anywhere within the general interaction region will be ~ 1 mV/m in a 150 Hz bandwidth (150 Hz is the approximate bandwidth at which these measurements were made). This correlation is certainly not perfect but leads one to believe that most of the observed noise can be tied to this contaminant release.

In order to properly scale the data to Space Station, several parameters need to be known:

1) The mass ejection rate and composition of gas leaking from the cabin and released through vents.

2) The ionization rate of the gas.

For purposes of this paper, we shall take the level measured near the Shuttle resulting from the offgassed water as our upper bound. Figure 7, taken from Gurnett et al [1988], shows a typical spectrum of this noise measured several hundred meters from the Orbiter. As can be seen, it is pseudo broadband below about 10^4 Hz and Gurnett et al [1988] indicate its wavelength is ≤ 1 meter. Clearly, this noise can be minimized by assuming that vents or thrusters are not operated during quiescent periods and that seals on pressurized modules have leak rates commensurate with the EME requirements.

8.0 EMISSION FROM ELECTRON BEAMS

The use of electron beams to study the phenomena associated with naturally occurring beams in the auroral region has a rich history in ground and flight experiments as well as in theoretical studies and computer simulation. It is not the purpose of this paper to review this work in any detail. The reader should consult the references for more information. Here we shall draw on data from experiments flown on rockets and the Shuttle to estimate the kind of electrical interference that may be expected when such experiments are conducted.

Beams emitting DC current and pulsed current have been investigated with energies ranging from ~50 ev [Koons et al, 1982] to 8 kev [Beghin et al, 1984] and currents less than 1 ma to several hundred milliamps. A wide range of plasma wave types have been observed. Typically, emission at the electron gyrofrequency and plasma frequency has been observed as well as ion and electron whistler waves [Shawhan et al 1984; Neubert et al, 1986; Reeves et al, 1988; Winckler et al, 1985]. Sources of these waves, which serve to scatter the beam and convert some of its kinetic energy into electromagnetic energy have been studied extensively. Farrell et al [1988] and Okuda et al [1988] are excellent sources for this topic. We are concerned here with the final result -- that is, what are the expected field strengths measured by an observer close to the experiment? For the answer we turn to measurements made on three specific Shuttle missions: OSS1, Spacelab 1, and Spacelab 2.

The wave emission depends on the injection pitch angle relative to the magnetic field [Neubert et al, 1986] and, to a certain degree, on the current and energy of the beam. In addition to narrowband emission at the gyro frequency and plasma frequency,

strong waves are always observed in the VLF band between about 750 Hz and 10 kHz with an f^{-n} spectral density where n varies from ~.7 to 1.5 [Farrell et al, 1988]. Detailed classifications of the spectra have been carried out by Akai [1984]; and Shawhan et al [1984]. We shall use the results from Shawhan et al [1984], Neubert et al [1986], and Reeves et al [1988] to place an envelope on the narrowband electric and magnetic emission. Figures 8a and 8b illustrate the probable upper bound of these emissions assuming a beam current of ~100 ma and a beam energy of 1-5 kev.

Although not directly related to EMI, the issue of charge balance for the Space Station must also be addressed. A recent, two dimensional simulation of this problem by Okuda and Berchem [1988] notes that charging can take place to fairly high potentials during beam operation. This charging is not a problem in itself but its consequences must be studied on a case by case basis. No significant charging was observed on Spacelab when the engine nozzles had access to the ambient plasma. However, charging was observed on Spacelab at comparable beam currents when the engine nozzles were in the wake. (The nozzles con- tribute ~30 m^2 to the conducting surface area of the Shuttle Orbiter and are the primary current return path.)

Keeping the prime conducting area of Space Station near the center of the vehicle and assuring a collecting area $\geq$100 m^2 should accommodate beam currents of several hundred milliamps with charging measured only in 10's of volts. Large current beams (>1 amp) and those with energies greater than a few kilovolts should provide, as part of their experiment, a system to insure charge neutralization. Detailed analysis can be undertaken once such an experiment and the Space Station conductive structure have been defined.

9.0 SUMMARY

Table II presents a summary of the wave source, wave type, and probable frequency ranges based on this review. To minimize sources of EMI from Space Station/ environment interactions, the following actions are recommended.

1) minimize leakage of 20 kHz and harmonic currents to structure by careful design of converters, interfaces, and cable; assure that the current return path does not include the structure but is carried along the 'green wire' to minimize loop area;

2) Study the effects of sheath waves on the propagation of 20 kHz and harmonics as these waves may raise levels of electric and magnetic noise due to leakage currents by several orders of magnitude;

3) minimize contact with the background plasma by making surfaces (e.g. solar arrays, cable trays, etc.) non-conductive; contact with the plasma should be made at one 'point' or area near the center of the station to avoid large v x B potentials (at least 100 m^2 is appropriate);

4) conduct design studies and laboratory tests under realistic flight conditions to assure that solar arrays can be operated at voltages which do not arc;

5) determine by analysis and test the effect of debris and micrometeroid impact holes on the arc rate of the solar arrays;

6) pay careful attention to the location and look direction of sensors sensitive to DC or low frequency magnetic fields and electric fields from wakes; consider that ASTROMAG operations may need to be scheduled carefully and that long term operation of the magnet may preclude certain other experiments;

7) analyze the ionization of gas leakage and vent products to determine if the broadband emission environmental requirements can be met during quiescent periods; develop a model which incorporates ionization rates, plasma dynamics, and neutral gas dynamics;

8) implement all of the following methods to minimize gaseous contamination which may ultimately affect the EME (this will also affect surface deposition):

a) The Orbiter should be allowed to outgas for $\geq$24 hours before docking with the Station (the Orbiter should be behind the Station).

b) Procedures minimizing thruster activity and plume impingement should be implemented for docking activity.

c) Any plan which includes continuous thrusting for reboost is eliminated based on EME considerations. The noise environment would exceed the specification by several orders of magnitude if the product of the thruster exhaust exceeds $\geq$.1 g/s of H_2O.

d) Brief gaseous releases, either by Station hardware or other equipment, must be minimized, documented, and made available to users on a common data buss.

e) EVA activity should be confined to non-quiescent periods whenever possible. (This assumes a vented suit.)

9) Many investigations sensitive to background noise level, may not be able to schedule simultaneous operation with an electron beam experiment. Experiments that produce beams of ~1 amp of current should provide an additional source of neutralization.

"The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration."

REFERENCES

1. Acuna, M.H., N.F. Ness, "Results from the GSFC Fluxgate Magnetometer on Pioneer II", in Jupiter, ed. by T. Gehrels, 1976, p.830.

2. Adai, K., "Electron Beam-Plasma Interactions Experiment in Space", Res. note 285, Inst. of Space and Astronaut. Science, Tokoyo, 1984.

3. Banks, P.M., P.R. Williamson, and W.J. Raitt, "Space Shuttle Glow Observations", Geophysics Res. Lett., 10, p.118, 1983.

4. Barnett, A., S. Olbert, "Radiation of Plasma Waves by a Conducting Body Moving Through a Magnetized Plasma", J. Geophys. Res., p.10, 117, Sept., 1986.

5. Beghin, C., J.P. Lebreton, B.N. Maehlum, J. Troim, P. Ingsoy, and J.L. Michau, "Phenomena Induced Charged Particle Beams", Science, 225, p.188, 1984.

6. Bernhardt, P.A., B.A. Kashiwa, C.A. Tepley, and S.T. Noble, "Spacelab 2 Upper Atmospheric Modification Experiment Over Arecibo, 1, Neutral Gas Dynamics", Astro. Lett. and Comm., 27, p.169, 1988a.

7. Bernhardt, P.A., W.E. Swartz, M.C. Kelly, M.P. Selzer, and S.T. Noble, "Spacelab 2 Upper Atmospheric Modification Experiment Over Arecibo, 2, Plasma Dynamics", Astro. Lett. and Comm., 27, p.183, 1988b.

8. Drell, S.D., H.M. Foley, M.A. Ruderman, "Drag and Propulsion of Large Satellites in the Ionosphere: an Alfven Propulsion Engine in Space", J. Geophys. Res., p.3131, July,1965.

9. Farrell, W.M., D.A. Gurnett, P.M. Banks, R.I. Bush, W.J. Raitt, "An Analysis of Whistler Mode Radiation from the Spacelab 2 Electron Beam", J. Geophys. Res., p.153, Jan., 1988.

10. Ferguson, Dale C., "The Voltage Threshold for Arcing for Solar Cells in LEO: Flight and Ground Test Results", NASA TM 87259, 1986.

11. Grebowsky, J.M., H.A. Taylor, Jr., M.U. Pharo III, and N. Reese, "Thermal Ion Perturbations Observed in the Vicinity of the Space Shuttle", Planetary Space Sci., 35, p.501, 1987.

12. Grier, N.T., "Plasma Interaction Experiment (Pix II) Laboratory and Flight Results", Spacecraft Environments Interactions Technology Conference, NASA #2359/AFGL-TR-85-0018, 1983.

13. Gurnett, D.A., W.S. Kurth, J.T. Steinberg, and S.D. Shawhan, "Plasma Wave Turbulence Around the Shuttle: Results from the Spacelab-2 Flight", Geophys. Res. Lett, 15, p.760, 1988.

14. Hansen, I.G., "EMC and Power Quality Standards for 20 kHz Power Distribution", Proc. of IECEC, p.356, 1987.

15. Hastings, D.E., G. Weyl, and D. Kaufman, "A Simple Model for the Threshold Voltage for Arcing on Nagatively Biased High Voltage Solar Arrays", Journal of Spacecraft and Rockets, submitted 1989.

16. Hastings, D.E., A. Barnett, S. Olbert, "Radiation from Large Space Structures in Low Earth Orbit with Induced Alternating Currents", J. Geophys. Res., p.1945, March, 1988.

17. Hastings, D.E., and J. Wang, "Induced Emission of Radiation from a Large Space Station Like Structure in the Ionosphere", AIAA Journal, April, 1989.

18. Hunton, D.E. and J.M. Calo, "Low energy ions in the Shuttle environment: Evidence for Strong Ambient-Contaminant Inter-Actions", Planetary Space Sci., 33, p.8, 1985.

19. Hunton, D.E. and W. Swider, "Variations of Water Vapor Concentration in the Shuttle Environment", Journal of Spacecraft and Rockets, 25, p.139, March-April, 1988.

20. Hwang, K.S., M.H. Stone, K.H. Wright, Jr., and U. Samir, "The Emissions of Broadband Electrostatic Noise in the Near Vicinity of the Shuttle Orbiter", Planet Space Sci., 35, p.1373, 1987.

21. Jongeward, G.A., I. Katz, M.J. Mandell, and D.E. Parks, "The Role of Unneutralized Surface Ions in Negative Potential Arcing", IEEE TNS vol. NS-32, Dec. 1985.

22. Koch, D.G., G.G. Fazio, W. Hoffman, G. Milnick, G. Rieke, J. Simpson, F., Witteborn, and E. Young, "Infrared Observations

of Contaminants from Shuttle Flight 51-F", Adv. Space Res., 7, (5) p.211, 1987.

23. Koons, H.C. and H.A. Cohen, "Plasma Waves and Electrical Discharges Simulated by Beam Operations on a High Altitude Satellite", in Artificial Partical Beams in Space Plasma Studies, monograph, edited by B. Grandal, p.111, Plenum, New York, 1982.

24. Kuninaka, H., and K. Kuriki, "Interference of High Voltage Solar Array with Ionospheric Plasma", 15th International symposium on Space Technology and Science Proceedings, Vol I, p.819, May, 1986.

25. Laurin, J.J., G.A. Morin, and K.G. Balmain, "Sheath Wave Propagation in a Magnetophasma", Radio Science, May-June, 1989.

26. Leung, P.L., "Discharge Characterisitcs of a Simulated Solar Cell Array", IEEE TNS vol. NS-30, p.4311, 1983.

27. Ma, T.Z., D.A. Garnett, and C.K. Goertz, "Interpretation of Electrostatic Noise Observed by Voyager I in Titan's Wake", J. Geophys. Res., p.8598, Aug., 1987.

28. McMahan, W., R. Salter, R. Hills, and D. Delorey, "Measured Electron Contribution to Shuttle Plasma Environment", AIAA-83-2598, AIAA Shuttle Environemnt and Operations Meeting, October 31-November 3, 1983.

29. Miller, E.R., (ed.), "STS-2,-3,-4 Induced Environment contamination Monitor (IECM) Summary Report", NASA TM-82524, 1983.

30. Miller, W.L., "An Investigation of Arc Discharges on Negatively Biased Dielectric-Conductor Samples in a Plasma", Spacecraft Environment Interactions Technology Conference, 1983, NASA #2359, AFGL-TR-85-0018.

31. Murphy, G.B., "Contaminant Ions and Waves in the Space Station Environment", NASA CP-3002, Proc. of Space Station Contamination Workshop, p.19, 1988.

32. Narcisi, R.S., "Quantitative Determination of the Outgassing Water Vapor Concentrations Surrounding Space Vehicles from Ion Mass Spectrometer Measurements", Adv. Space Res., 2, p.10, 1983.

33. Neubert, T., W.W.L. Taylor, L.R.O. Storey, N. Kawashima, W.T. Roberts, D.L. Reasoner, R.L. Williams, and J. Burch, "Waves Generated During Electron Beam Emissions from the Space Shuttle", J. Geophys. Res., 91, p.321, 1986.

34. Okuda, H., and M. Ashour-Abdalla, "Ion Accoustic Instabilities Excited by Injection of an Electron Beam in Space", J. Geophys. Res., 93, p.2011, March, 1988.

35. Okuda, H., J. Berchem, "Injection and Propagation of a Nonrelativistic Eelectron Beam and Spacecraft Charging", J. Geophys. Res. 93, p.175, Jan. 1988.

36. Paterson, W.R., and L.A. Frank, "Hot Ion Plasmas from the Cloud of Neutral Gases Surrounding the Space Shuttle", J. Geophys. Res. 94, p.3721, April, 1989.

37. Pickett, J.S., G.B. Murphy, and W.S. Kurth, "Gaseous Environment of the Shuttle Early in the Spacelab 2 Mission", Journal of Spacecraft and Rockets, 25, p.169, March-April, 1988.

38. Pistole, C.O., "Impact of Power Distribution on the Space Station EMI Environment", SAE/P-85/164.

39. Purvis, C.K., "The PIX II Experiment: An Overview", Spacecraft Environmental Technology Conference, NASA #2359 AFGL-TR-85-0018, 1983.

40. Reeves, G.D., P.M. Banks, A.C. Fraser-Smith, T. Neubert, and R.I. Bush, "VLF Wave Stimulation by Pulsed Electron Beams Injected from the Space Shuttle", J. Geophys. Res., 93, p.162, 1988.

41. Renz, D.D., R.C. Finke, N.J. Stevens, J.E. Triner, and I.G. Hansen, "Design Considerations for Large Space Electric Power Systems", NASA TM-83064, April, 1983.

42. Rice, R.R., "Space Station Power System Selection", Proc. of 21st IECEC, Vol. 3, p.1886, Aug., 1986.

43. Schmitz, G.V., and J.J. Biess, "Power Transmission Cable Development for the Space Station Freedom Electrical Power System", Proc. IECEC, 1989.

44. Shawhan, S.D, G.B. Murphy, and D.L. Fortna, Measurements of Electromagnetic Interference on OV102 Columbia Using the Plasma Diagnostic Package", J. Spacecraft and Rockets, 21, p.4, 1984a.

45. Shawhan, S.D., G.B. Murphy, and J.S. Pickett, "Plasma Diagnostics Package Initial Assessment of the Shuttle Orbiter Plasma Environment" J. Spacecraft & Rockets, 21, p.387, 1984b.

46. Shawhan, S.D., G.B. Murphy, P.M. Banks, P.R. Williamson, and W.J. Raitt, "Wave Emissions from DC and Modulated Electron Beams on STS-3", Radio Science, 19, p.471, March-April, 1984.

47. Simon, W.E., and D.L. Nored, "Manned Spacecraft Electrical Power Systems," Proc. IEEE, p.277, March, 1987.

48. Snyder, D.B., "Characteristics of Arc Currents on a Negatively Biased Solar Cell Array in a Plasma", IEEE TNS, Vol NS-31, pp.1584-1587, Dec., 1984.

49. Stevens, N.J., M.E. Kirkpatrick, G.K. Crawford, and D.P. Hansen, "High Voltage System Performance in Low Earth Orbit Plasma Environment", TRW 46870-6005-UT-00 NASA contract #NAS3-24659 Oct. 1986.

50. Sullivan, J.D., D.E. Hastings, and B.G. Lane, "Investigation of Electromagnetic Inter- Action Caused by the Operation of High Level Magnetic Field Experiments", Final Report JPL contract #958231, Jan., 1989.

51. Torr, M.R., "Optical emissions induced by Spacecraft-Atmosphere Interactions", Geophys. Res. Lett., 10, p.114, 1983.

52. Torr, M.R., and D.G. Torr, A Preliminary Spectroscopic Assessment of the Spacelab 1/ Shuttle Optical Environment", J. Geophys. Res., 90, p.1683, 1985.

53. Torr, M.R., D.G. Torr, and J.K. Owens, "Optical Environment of the Spacelab 1 Mission", Journal of Spacecraft and Rockets, 25, March-April, 1988.

54. Tribble, A.C., J.S. Pickett, N. D'Angelo, and G.B. Murphy, "Plasma Density Temperature, and Turbulence in the Wake of the Shuttle Orbiter", in Press, Planetary Space Science, 1989.

55. Winckler, J.R., K.E. Erickson, Y. Abe, J.E. Steffen, and P.R. Malcolm, "ELF Wave Production by an Electron Beam Emitting Rocket System", Geophys. Res. Lett., 12, p.457, 1985.

56. Witteborn, F.C., L. O'Brient, and L. Caroff, "Measurements of the Nighttime Infrared Luminosity of Spacelab 1 in the H and K Bands", NASA TM-85972, 1985.

57. Wulf, E., and U. Von Zahn, "The Shuttle Environment: Effects of Thruster Firings on Gas Density and Composition in the Payload Bay", J. Geophys. Res., 91, pp.3270-3278, 1986.

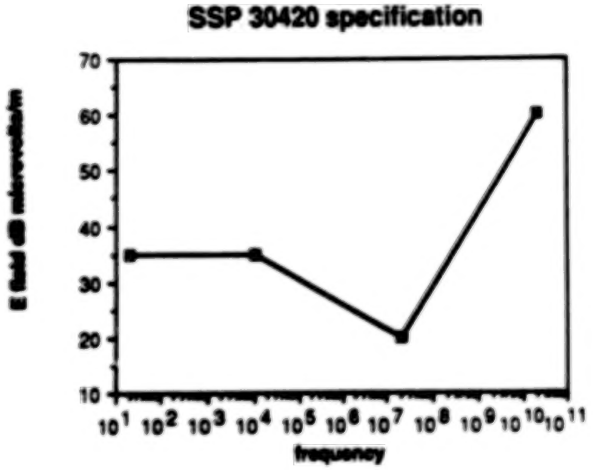


Figure 1a.

The narrowband electric field environment specified 1 meter from structure. (SSP 30420 Oct. 1, 1986).

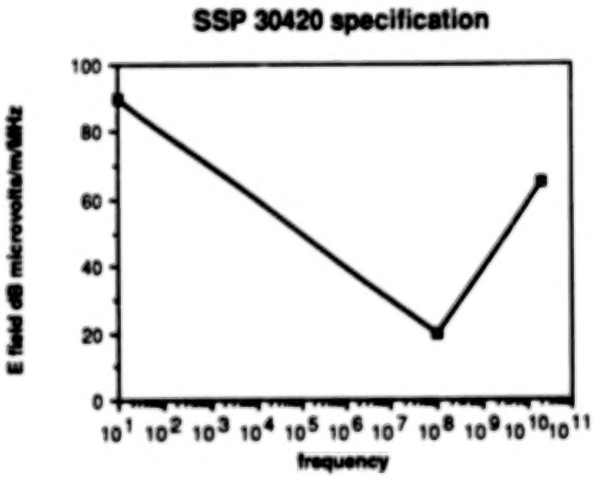


Figure 1b.

The broadband electric field environment at 1 meter from structure. (SSP 30420 Oct. 1, 1986).

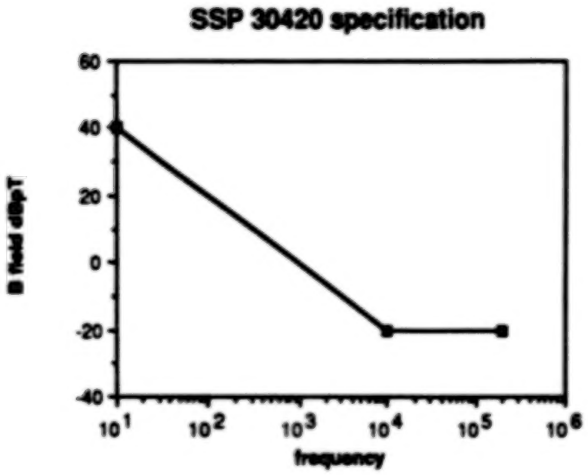


Figure 2.

The narrowband magnetic field environment (SSP 030420 Oct 1, 1986). SSP 30237 indicates a ±10 kHz wide notch around 20 kHz to allow for the 20 kHz power supply noise.

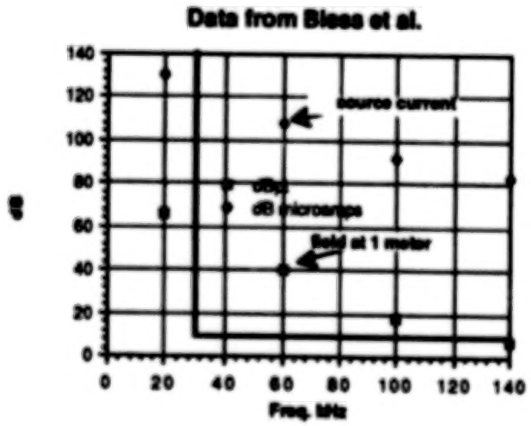


Figure 3a.

The radiated magnetic field of the transmission extrapolated to 1 meter from measurements by Biess et al. Note that the transmission line current at each frequency is also shown.

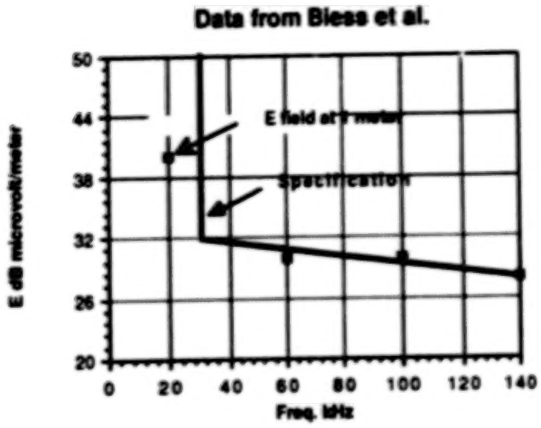


Figure 3b.

The radiated electric field of the transmission line at 1 meter (Biess et al).

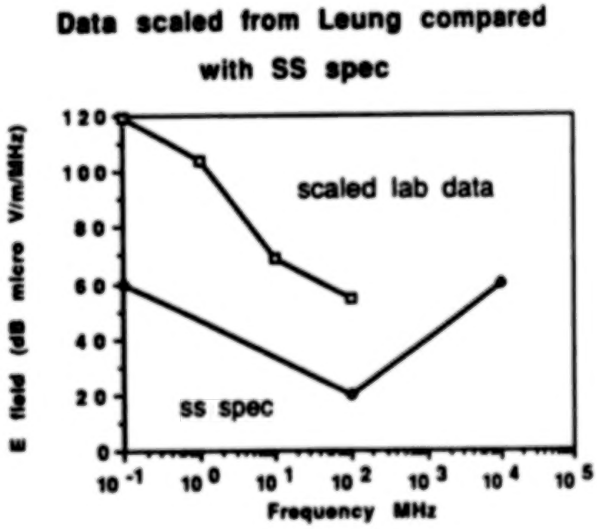


Figure 4.

The probable magnitude of the broadband electric field generated by arcs on the solar array. Measurement distance is assumed to be 20 meters. Data are from Leung [1984].

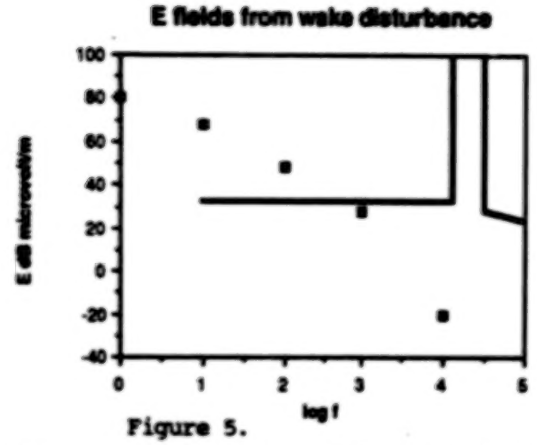


Figure 5.

The probable magnitude of electric field wake noise near the mach cone of an object of characteristic dimension ~10m.

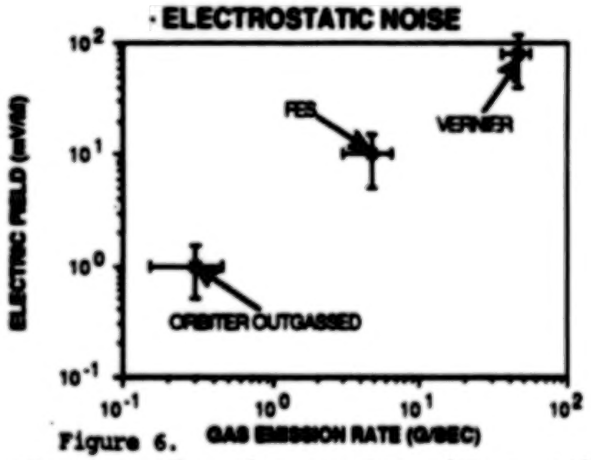


Figure 6.

Electrostatic noise at 1kHz with a ±15% bandwidth is shown for three different levels of gas emission (derived from Pickett et al [1984]).

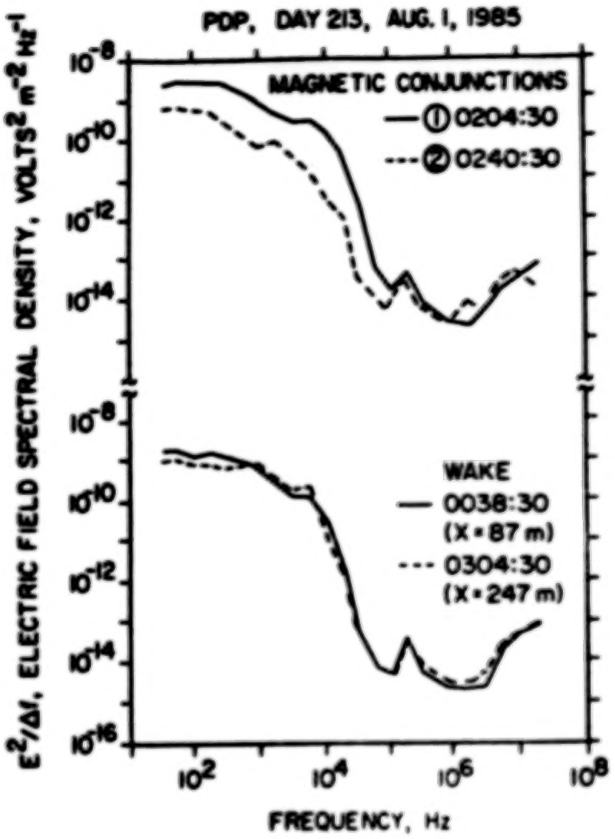


Figure 7.

Graph is from Gurnett et al [1988] and indicates the electric spectrum of noise induced by the ionization of gas clouds (top) and that induced by wakes (bottom).

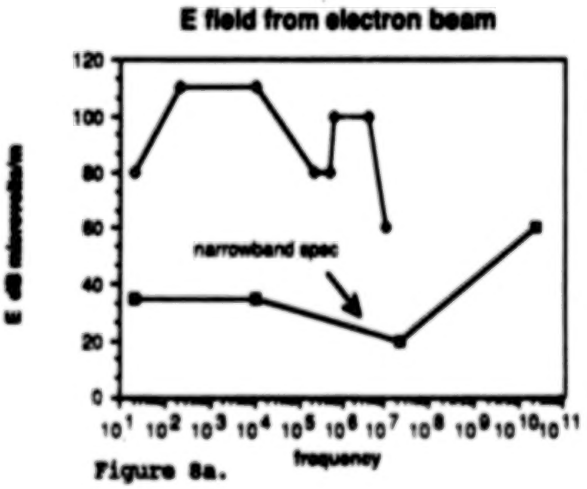


Figure 8a.

The envelope of probable electric field emissions due to a 100ma 1kev electron gun at a distance of ~5 meters. Also shown is the narrowband emission limit of figure 1a.

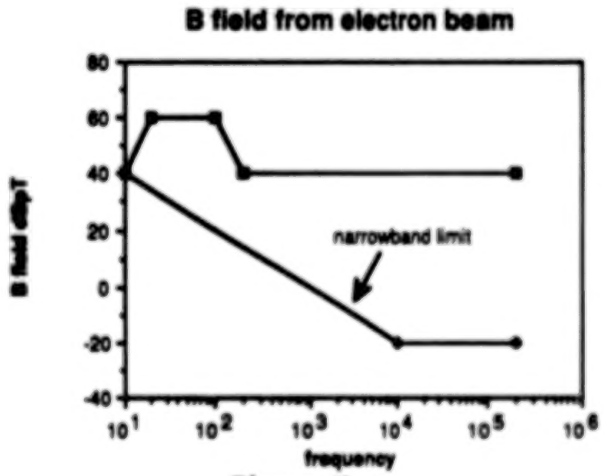


Figure 8b.

The envelope of magnetic field noise induced by the same electron beam is compared to the SSP 30420 specification.

Table I

radiation type	frequency range	source
lower hybrid waves + cyclotron waves	40Hz - 10kHz broadband 40Hz + others	ring distribution ions - primarily the wake region
plasma turbulence	broadband 10Hz < f≤ 6 kHz	plasma wake region
lower hybrid waves	100 - 200 kHz	boundary layer at ion turning pt.
upper hybrid waves	30 MHz	boundary layer
whistler waves	100 kHz - 1 GHz	loss cone (cusp regions)
Alfven waves	stationary but time varying amplitude	generated by ionospheric current collection

Table II

Source	Wave type	Wave Classification: Electric	Wave Classification: Magnetic	Frequency Range	Magnitude Estimate (at 1 meter): Electric	Magnetic
Power System						
transmission line	electromagnetic	E	B	20 kHz and harmonics	~60 dBμV/m* at 20 MHz	≤70 dBγ* at 20 kHz
current loops	sheath wave lower hybrid	E E	B	20 kHz and harmonics	100 dBμV/m* at 20 kHz	140 dBγ* at 20 kHz
Ionospheric Current	Alfvén wave lower hybrid wave	PB	B	DC - 100 Hz 20 kHz and harmonics	unknown is Doppler shifted near field	≤74 dBγ
ASTRONAS	DC magnetic (see Table I)		B	DC <F<100 Hz		>60 dBγ**
ABCS	electromagnetic	E	B	100 kHz<f<100 MHz	110 dBμV/m/MHz at 1 MHz	E = B/120*
VLBI	electrostatic	PB		0<f<10 kHz	≤50 dBμV/m at 100 Hz	
Gas Cloud	electrostatic	DC		0<f<10 kHz at 1 kHz	≤60 dBμV/m**	
Electron Beam	electron gyrofreq. plasma frequency electron whistlers electrostatic ion gyrofrequency	E E PB DC E	B PB B	~1 μHz ~3 μHz 10 kHz<f<1 MHz 10 Hz<f<10 kHz	≤100 dBμV/m ≤100 dBμV/m ≤100 dBμV/m ≤110 dBμV/m ≤120 dBμV/m	unknown unknown ≤60 dBγ ≤60 dBγ ≤60 dBγ

* based on prototype inverter and transmission line
\+ assumes 1 ma leakage current; 2m x 50m loop; 10cm sheath
** 60dBγ at 75 meters
** assumes 1g/s water emission rate

SOLAR ARRAY ARCING IN PLASMAS

Dale C. Ferguson
Space Environment Effects Branch
NASA Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135

ABSTRACT

Solar cells in space plasma conditions are known to arc into the plasma when the interconnects are at a negative potential of a few hundred volts, relative to plasma potential. For cells with silver-coated interconnects, a threshold voltage for arcing exists at about -230 V, as found in both ground and LEO experiments. The arc rate beyond the threshold voltage depends nearly linearly on plasma density, but has a strong power-law dependence on voltage, such that for small increments in operating voltage there is a large increment in arc rate. The arcs generate broadband radio interference and visible light. In ground tests, interconnects have been damaged by arcs in cells having insufficient isolation from a source of high current. Models for the arcs are highly dependent on the choice of interconnect conductor material exposed to the plasma and possibly on the geometry and choice of adjacent insulator material. Finally, new technology solar cells use copper for the cell interconnects, a material which may have a lower arcing threshold voltage than silver. It is expected, from ground tests of simulated solar cells, that any junction of conductor and insulator exposed to space plasma conditions will arc into the plasma at a few hundred volts negative potential, relative to the local plasma.

INTRODUCTION

The prospect of flying large structures in space brings with it a need for space power systems capable of generating large amounts of power. To keep cable masses low, with no loss of efficiency, high voltages must be used; much higher than the 28 V systems typically orbited until now, and even higher than the occasional 100 V used on Skylab.

Solar cells, which individually generate low voltages, are typically strung together in series for high power applications, so that the total voltage across the array may become quite large. The connections from one cell to another are called interconnects. In standard technology solar arrays, the interconnects are coated with silver, for ease in soldering, and are exposed to the surrounding environment. Newer technology cells are welded to a copper trace from the back, so that little conductor is exposed in front. If the cell backs are not well insulated, copper will contact the space plasma.

Early plasma testing of solar cells in simulated low Earth orbit (LEO) plasmas (Cole et al 1968, Stevens 1978) revealed that at high positive array potentials of a few hundred volts relative to the plasma, even the insulating cover glasses collected electrons from the plasma as if they were conductors. This effect, known as "snapover", has been understood in terms of secondary electrons generated on the cover glasses "hopping" over to be collected at the conductor.

At high negative potentials of a few hundred volts relative to the plasma, solar cells were observed to arc into the surrounding plasma, first in ground tests (Kennerud 1974) and later in orbital flight tests (PIX-I and PIX-II, Grier and Stevens 1978, and Grier 1983). PIX-I, because of limited plasma diagnostics, essentially only confirmed that arcing was not an effect caused by the plasma chamber walls. PIX-II, however, yielded information about arcing voltage thresholds and arc rates, as well as about the "snapover" electron currents, in LEO conditions.

Because of the obvious implications of arcing and anomalous current collection on systems exposed to the space plasma, it is of some interest to understand these plasma interactions with spacecraft systems. This paper reviews current progress in understanding solar array arcing in plasmas.

ARC RATE AND THRESHOLD MEASUREMENTS

Ferguson (1986) shows that the onset of arcing in solar array plasma tests may not accurately reflect the voltage threshold. If, for example, the arc rate at some combination of plasma conditions and bias voltage is very low, the experimenter may move on to higher voltages before arcing is observed. When arcs were observed in ground tests, and arc rates could be obtained (Miller 1983, Leung 1985, and Grier 1984), it was found that the arc rate depended on the conditions in the following approximate way:

$$R = C_1 \, n \, T^{0.5} \, m^{-0.5} \, V^x, \quad (1)$$

where C_1 is a constant, n is the plasma density, T is the plasma temperature, m is the plasma ion mass, V is the interconnect voltage relative to the plasma potential, and x is approximately equal to 5 for 2x2 cm cells, 2x4 cm cells, and the fronts of 5.9x5.9 cm cells in ground tests, and x is about 8 for the fronts and backs of 5.9x5.9 cm cells together, in ground tests. The PIX-II flight results yielded a value for x of about 3 for 2x4 cm cells in orbit. The difference between the fronts of 5.9x5.9 cm cells only and the fronts and backs together may be caused by a difference in the exposed conducting materials on the cell fronts and backs as will be discussed later. The difference between x for the 2x4 cm cells in ground tests and x in space may be due to the presence of atomic oxygen in space, as contrasted with other gases used in ground tests.

If the voltage at which arcing is first observed in a test is interpreted as that voltage at which the average time interval between arcs becomes less than the continuous test time at that voltage, it may be shown that (Ferguson, 1986):

$$V_{on} = C_2 \, n^{(-1/x)}, \quad (2)$$

where V_{on} is the apparent onset voltage, C_2 is a constant, and n and x are as defined before. Thus, an apparent density dependence of the arcing threshold may in fact be simply a reflection of the steep voltage dependence of the arc rate.

The true voltage threshold for arcing for 2x4 cm cells with silver-coated interconnects, defined as the potential below which the measured arc rate is several standard deviations below the rate extrapolated from higher voltages, has been found to be about -230 V, relative to the plasma, from all available ground and orbital data. This threshold may be a function of solar cell geometry and materials, and should not be taken to represent the threshold for arbitrary or new solar array designs. Figure 1 shows the arc rate behavior found for several arrays of 2x2 cm and 2x4 cm cells in ground and orbital tests, normalized by the plasma parameters in equation (1). It is worth mentioning that the arc rate observed does not depend strongly on the number of cells or exposed interconnects at high voltage. For a large array with insulated interconnects, one pinhole in the insulation will thus arc effectively as much as if the interconnects were all exposed to the space plasma.

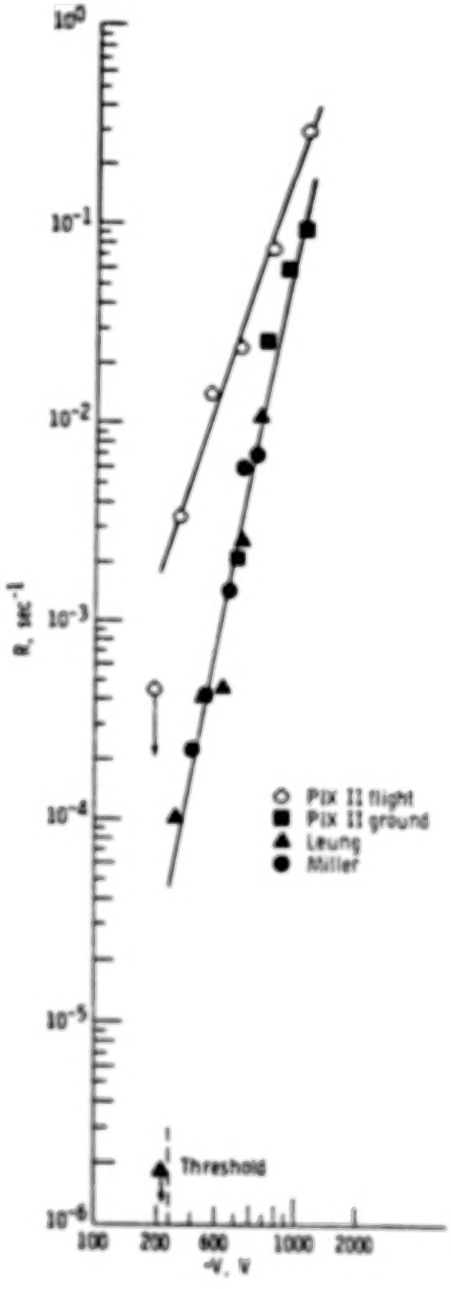


Figure 1 - Arc rate versus voltage for standard interconnect cells, normalized to LEO ram conditions.

EFFECTS OF SOLAR ARRAY ARCS

Of course, it is only necessary to avoid arc occurrence if the arcs may harm spacecraft operations or systems. Leung (1985), as part of the now-defunct VOLT experiment development, measured the radiofrequency noise spectrum of arcing solar cells. Figure 2 shows the spectrum he found. Depending on the strength of the arcs (which depends on the capacitance of the array to space, as also found by Snyder in 1985), the EMI generated may be negligible or quite significant. For arrays large enough to generate the high potentials necessary for arcing, EMI may be significant if it couples to spacecraft electrical systems. Communications between spacecraft, or between spacecraft and telerobotic systems may be disrupted.

In addition to radiofrequency EMI, arcs produce visible light, which may interfere with optical experiments. The visible spectrum of the arcs has not, to date, been measured.

If arcs occur in solar arrays which are insufficiently isolated from a high current source, as in early experiments where a high voltage power supply was used to bias the arrays, the large arc currents may damage materials at the arcing point. Miller (1983) found partially melted interconnect material in arrays which had been repeatedly arced. Since his experience, it has become standard experimental technique to place a large resistance between bias sources and the array to be plasma tested. For large space solar arrays, however, the source of the high negative potentials may be the array itself. In this case, the strength of the arcs will depend on the total array capacitance and on internal array connections (diodes, etc.).

Adverse array arcing effects may be mitigated in several ways:

1. Design the system so that high negative potentials relative to the plasma will exist nowhere in the system. This will mean, in practice, one of two design solutions. Either the total array voltage, from end to end, must be limited, or a large current-collecting area is provided at the negative end of the array, so

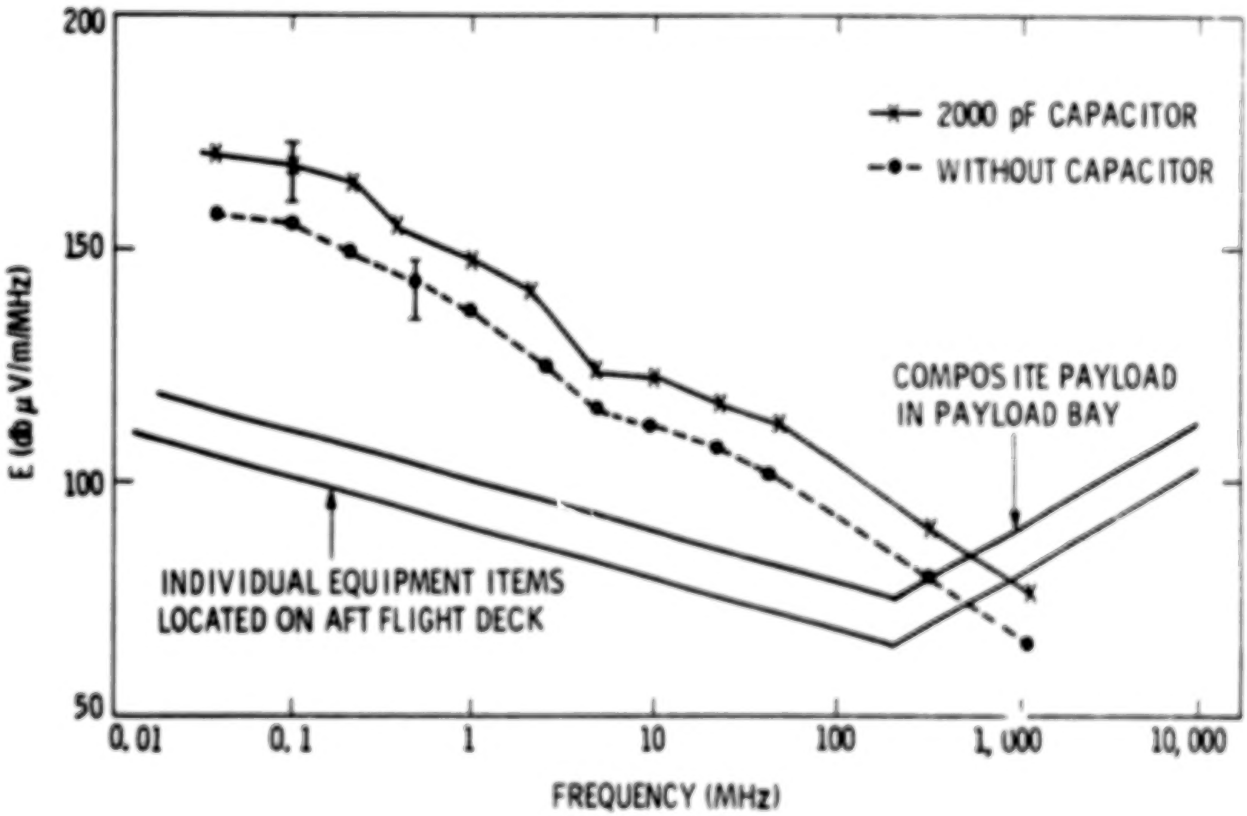


Figure 2 - Spectra of RF radiation generated by a 4 cell array of 5.9x5.9 cm cells, with welded-through interconnects, with and without an added capacitance.

there will be a low potential drop at that end. The first solution will mean that larger cable masses must be used to limit I^2R losses. The second solution will push the positive end of the array up to high positive potentials relative to the plasma, so that snapover problems may become important.

2. Insulate the interconnects so thoroughly that it is certain no plasma contact will be made. For long-lived arrays, micrometeoroids and/or debris may puncture the insulation, nullifying this solution.

3. Design the arrays so that only a small amount of charge may be dumped when an arc occurs. This may limit the EMI, etc., to tolerable levels. Bear in mind, though, that each independent array segment will arc as one unit, so the total arc rate will go up linearly as the number of arc segments.

4. Use materials for interconnects, coatings, etc., which will not arc at the high negative potentials likely to exist on parts of the array. This option will be further examined in the next section.

POSSIBLE MATERIAL DEPENDENCES

Interesting differences exist in arcing thresholds and arc rates between arrays having different surface materials. For instance, ESA tests (Thiemann and Bogus 1986 and 1988) have shown that arrays having the interconnects coated with an "insulating" material sometimes arced at lower voltages than they did before the interconnects were coated. The effect amounted to hundreds of volts in the onset of arcing.

Solar arrays for the PIX-II experiment had a much higher voltage for the onset of arcing in pre-flight ground test experiments than they exhibited in orbit (Ferguson, 1986). The voltage dependence of the arc rate of the PIX-II arrays was also very different than found in pre-flight testing. In this case, the orbital arrays were likely covered with a thin oxide coating from interaction with the atomic oxygen in LEO, whereas pre-flight testing was done in an argon plasma.

Miller (1983) showed that in ground tests the solar array arc rate started out high, and then decreased to a constant level on a time scale of a few hours, as if the arc sites were cleaning themselves up during testing. Leung (1985) observed the same effect in ground tests using a different background plasma. PIX-II also showed this effect in orbit (Ferguson 1986).

As was already mentioned, the fronts of 5.9x5.9 cm cells arced in ground tests at a slower rate than the fronts and backs together. Here, the significant fact may be that the cell backs had copper exposed to the plasma, rather than silver. Supporting this contention are the ground tests performed by Snyder (1986) on metals partially covered with insulating material, to simulate the conductor-insulator junctions on solar cells. Silver in his tests arced so as to bring the local conductor potential down to about a -230 V level, coincident with the arcing threshold found in other tests. Copper in his tests arced so as to bring the local copper potential in arcs to a value less than about -120 V, possibly indicating a lower arcing threshold voltage for copper than for silver.

All of these results may be understood if the arcing threshold and arc rate depend on the surface properties of the materials at the arc sites. There are two popular models for the onset of arcing at high negative potentials. In one, a thin dielectric layer of contaminant is built up on the surface of the conductor, and sufficiently high electric fields may be produced in the vicinity of the insulator to punch through the layer, triggering an arc (Jongeward et al 1985). In the second model, breakdown of gas emitted by the insulator under electron bombardment may lead to an avalanche into the plasma if the electric fields are high enough (Hastings et al 1989). In both of these models material properties play a strong role, as does the presence of high electric fields near conductor-insulator interfaces. On the basis of these models and the observations they are meant to explain, arcs may be expected at negative potentials of a few hundred volts negative, relative to the plasma, at conductor-insulator junctions, regardless of whether the junction occurs on a solar cell or is part of some other spacecraft system.

Obviously, more work needs to be done in ground tests and space flight experiments, to investigate the material and geometry dependences of the arc rate and arc threshold. Only then can proper mitigation techniques be employed. One approved Shuttle experiment to investigate arcing on solar arrays in LEO is the SAMPIE, or Solar Array Module Plasma Interaction Experiment, a joint NASA/ESA venture now manifested for late in 1994. Ground testing continues at NASA Lewis Research Center, TRW, and elsewhere.

CONCLUSIONS

EMI generated in solar array arcs may generate radiofrequency noise which might

disrupt telerobotic communications. High negative potentials on other spacecraft surfaces are a possible threat to the successful operation of spacecraft systems, including automation and robotics electronics and communications. It is necessary to consider solar array arcing and solar-array-type arcing in the design of spacecraft power systems and other systems which may be affected by arcing. Systems should be designed to mitigate the incidence and effects of arcing, whenever possible. Although not the topic of this paper, arcs may also occur during docking, if the potentials of the docking vehicles differ sufficiently. Thus, control of spacecraft potentials is important in spacecraft design, if reliability and communications are important to spacecraft systems. Material dependences of arc rates and thresholds are important factors in system design, and our knowledge of them relies to a great extent on lab and spaceflight experiments, some of which remain to be done.

REFERENCES

Cole, R.K., Ogawa, H.S., and Sellen, J.M., Jr. (1968), "Operation of Solar Cell Arrays in Dilute Streaming Plasmas", NASA CR-72376.

Ferguson, D.C. (1986), "The Voltage Threshold for Arcing for Solar Cells in LEO - Flight and Ground Test Results", NASA TM-87259.

Grier, N.T. (1983), "Plasma Interaction Experiment II (PIX II): Laboratory and Flight Results", Spacecraft Environmental Interactions Technology 1983, NASA CP-2359, pp. 333-347.

Grier, N.T. (1984), "Dilute Plasma Coupling Currents to a High Voltage Solar Array in Weak Magnetic Fields", 19th IECEC Conference, San Francisco.

Grier, N.T. and Stevens, N.J. (1978), "Plasma Interaction Experiment (PIX) Flight Results", Spacecraft Charging Technology 1978, NASA CP-2071, pp. 295-314.

Hastings, D.E., Weyl, G., and Kaufman, D. (1989), "A Simple Model for the Threshold Voltage for Arcing on Negatively Biased High Voltage Solar Arrays", Journal of Spacecraft and Rockets, submitted.

Jongeward, G.A. et al (1985), The Role of Unneutralized Surface Ions in Negative Potential Arcing", IEEE Trans. Nucl. Sci., vol. NS-32, no. 6, Dec., pp. 4087-4091.

Kennerud, K.L. (1974), "High Voltage Solar Array Experiments", NASA CR-121280.

Leung, P. (1985), "Characterization of EMI Generated by the Discharge of a 'Volt' Solar Array", JPL D-2644, Jet Propulsion Lab, California Institute of Technology, Sept. 1985.

Miller, W.L. (1983), "An Investigation of Arc Discharging on Negatively Biased Dielectric-Conductor Samples in a Plasma", Spacecraft Environmental Interactions Technology 1983, NASA CP-2359, pp. 367-377.

Snyder, D.B. (1985), "Characteristics of Arc Currents on a Negatively Biased Solar Cell Array in a Plasma", NASA TM-83728.

Snyder, D.B. (1986), private communication.

Stevens, N.J. (1978), "Interactions Between Spacecraft and the Charged-Particle Environment", Spacecraft Charging Technology 1978, NASA CP-2071, pp. 268-294.

Thiemann, H. and Bogus, K. (1986), "Anomalous Current Collection and Arcing of Solar-Cell Modules in a Simulated High-Density Low-Earth Orbit Plasma", ESA-Journal, Vol. 10, pp. 43-57.

Thiemann, H. and Bogus, K. (1988), "High Voltage Solar Cell Modules in Simulated Low-Earth-Orbit Plasma", Journal of Spacecraft and Rockets, Vol. 25, pp. 278-285.

PHOTOVOLTAIC ARRAY SPACE POWER PLUS DIAGNOSTICS EXPERIMENT

Donald A. Guidice
Geophysics Laboratory, Hanscom AFB, MA 01731

Abstract: The objective of the Photovoltaic Array Space Power Plus Diagnostics (PASP Plus) experiment is to measure the effects of the interaction of the low- to mid-altitude space environment on the performance of a diverse set of small solar-cell arrays (planar and concentrator, representative of present and future military technologies) under differing conditions of velocity-vector orientation and simulated (by biasing) high-voltage operation. Solar arrays to be tested include Si and GaAs planar arrays and several types of GaAs concentrator arrays. Diagnostics (a Langmuir probe and a pressure gauge) and a transient pulse monitor (to measure radiated and conducted EMI during arcing) will be used to determine the impact of the environment on array operation to help verify various interactions models. Direction of the effort is by AFSTC's Geophysics Lab and WRDC's Aero Propulsion and Power Lab, with Jet Propulsion Lab (JPL) as the experiment development contractor. Presently, JPL is finishing the assembly and testing of a brassboard unit; fabrication of a PASP Plus flight unit awaits the finding of a suitable spaceflight vehicle. Results from a successful PASP Plus flight will furnish answers to important interactions questions and provide inputs for design and test standards for photovoltaic space-power subsystems.

INTRODUCTION

Air Force mission requirements in the 1990s will necessitate larger, higher powered space systems. In supplying electrical power for such systems, consideration must be given to operating photovoltaic subsystems at higher voltage levels to reduce cable weight (minimize I^2R losses). New solar-cell materials are being investigated for higher efficiency. To make solar arrays less vulnerable to laser attack, various configurations are being investigated for "concentrator" arrays, which accept incoming energy from only a narrow angle around array boresight. These new technology innovations lead to new environmental interactions problems. To avoid launching space-power subsystems with built-in failure modes, environmental interactions questions must be answered before operational subsystems are built.

In 1985, AFSTC's Geophysics Laboratory (GL) and WRDC's Aero Propulsion and Power Laboratory (PO), through a Memorandum of Agreement, initiated the Photovoltaic Array Space Power (PASP) experiment development effort, with Jet Propulsion Laboratory (JPL) as the development contractor. Originally, PASP was to be one of several engineering technology experiments for GL's Interactions Measurement Payload for Shuttle (IMPS). However, after the Challenger loss, circumstances dictated that, instead of the full IMPS, we develop (and attempt to obtain spaceflights for) individual engineering technology experiments. After adding the necessary space-environment diagnostic sensors to PASP (these were originally part of IMPS), the experiment became PASP plus diagnostics, or PASP Plus for short.

PASP PLUS INSTRUMENTATION

The PASP Plus experiment consists of a set of solar-cell array modules, associated array-performance measurement equipment, and environmental diagnostic sensors.

PASP Plus can accommodate up to six solar array modules. At present we have four on hand; these are:

a. a silicon planar array [to be used as a standard] (see Figure 1).

b. an advanced-technology gallium-arsenide planar array (see Figure 2).

c. a mini-Cassegrainian concentrator array having eight small Cassegrainian-reflector structures (see Figure 3). The GaAs solar cell is located at the center of the base of the reflector surface. Two-dimensional concentration is achieved.

d. a SLATS (semi-parabolic low-aperture trough system) concentrator array, resembling Venetian blinds (see Figure 4). The concave curvature of a slat reflects incoming light onto a line of solar cells on the back surface of an adjacent slat. For SLATS, only one-axis concentration is achieved.

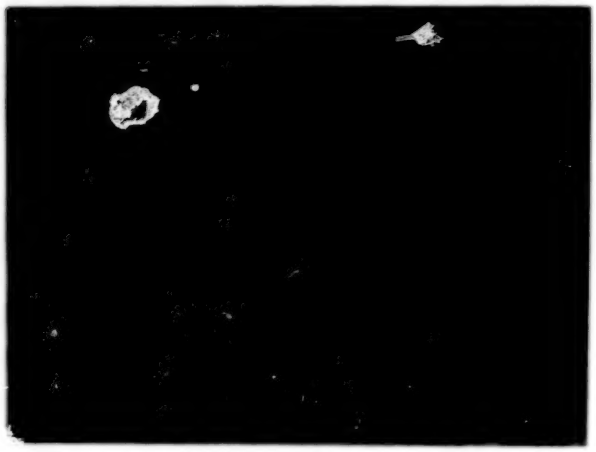


Figure 1. PASP Plus Silicon Planar Array Module.

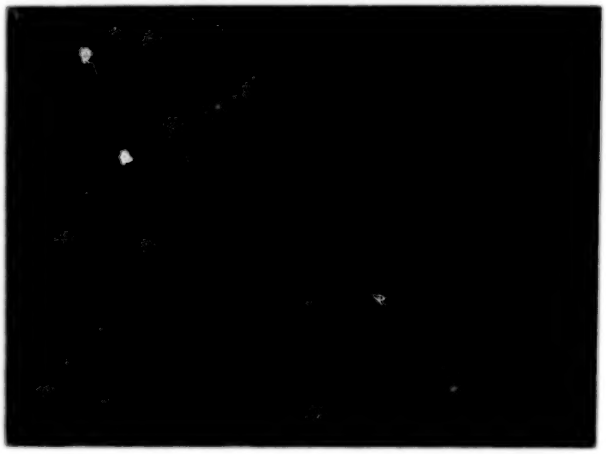


Figure 2. PASP Plus GaAs Planar Array Module.



Figure 3. PASP Plus Mini-Cassegrainian Concentrator Array Module.



Figure 4. PASP Plus SLATS Concentrator Array Module.

Instrumentation to measure array performance includes:

a. a sun incidence-angle sensor to measure the alignment of the array modules to the incident solar energy. This sensor is especially important for the concentrator arrays, which have only about a two-degree acceptable operating range around solar boresight. If no other means of sun-pointing is available on the carrier, the crossed-axes outputs from the sun incidence-angle sensor could be used to provide pointing information to the carrier.

b. direct current and voltage meters for I-V curve measurements.

c. temperature sensors on each array module, so array performance can be correlated with operating temperature.

d. electrical transient sensors (E-field for detection of radiated EMI and current-loop for detection of power-line EMI) connected to a transient pulse monitor (TPM) which will obtain the characteristics (amplitude, rise time, pulses per time period) of arc-discharge pulses that will occur during high-voltage biasing of the arrays.

Environmental diagnostic sensors of PASP plus include:

a. a pressure gauge to measure ambient pressures surrounding the arrays.

b. a Langmuir probe to measure low-energy plasma parameters (density and temperature).

Other functional assemblies include:

a. a multi-step high-voltage generator for biasing the arrays.

b. a buffer storage for intermediate data handling.

c. a controller network to control biasing and measurement sequences.

PASP PLUS OPERATIONS

The data gathering format for PASP Plus features a programmed sequence of measurements, with careful notation of ambient pressure [from pressure gauge], charged-particle density [from Langmuir probe], array temperature, array orientation toward the sun [from sun incidence-angle sensor], and array orientation with respect to vehicle velocity vector (ram, wake, in-between).

In a sequence for a given array, the current-voltage (I-V curves) measurements are made first. The high-voltage biasing operation follows next in the sequence. (The Langmuir probe is disconnected for the bias measurements.) Simulated high-voltage operation will be obtained utilizing eleven biasing steps from -500 to +500 Volts from the multi-step biasing generator. The particular values of biasing must be chosen beforehand (increments of ~25 V are available). Using preprogrammed bias values, PASP Plus progresses through the programmed sequence. The biasing is applied to one side of the array; each bias level will remain applied for 20 seconds. The 20-sec interval is chosen to have sufficient time to reach steady-state conditions (in the first few seconds) and still have enough time left in the interval to make statistically meaningful measurement of arc-discharge parameters (particularly, the number of arcs in the interval) with the TPM.

HIGH-VOLTAGE INTERACTIONS

Greater spacecraft power requirements bring about a need for higher-voltage power-distribution subsystems. Enhanced interactions between the space plasma and the arrays operating at higher voltages must be experimentally investigated in a systematic manner for different types of arrays. PASP Plus provides the means for accomplishing this task.

Based on the extensive work of the group at the NASA Lewis Research Center, important insights into high-voltage interactions in a space plasma (Ref. 1,2) have been obtained through both laboratory and flight test results (Ref. 3,4,5). Explanations concerning the arcing of negatively biased solar arrays have been put forth by Jongeward and Parks (Ref. 6,7); they suggest that arcing is initiated as a result of ion neutralization and associated charge buildup on a thin insulating layer over the metallic interconnects.

If one terminus of an array is operated at a high negative voltage (by biasing in the case of PASP Plus), the negatively biased part of the array will, beyond a certain voltage level, start to experience arcing. The arc discharges could damage the array (especially around the interconnects area) and generate unwanted electromagnetic interference that could cause false signals or even trigger erroneous commands.

If one terminus of an array is operated at a high positive voltage, the positively biased part will experience enhanced electron current collection. For an operational solar array, this effect would result in the loss of photovoltaicly generated power in the form of "leakage" current, diminishing the electrical power available to the spacecraft for useful (mission oriented) purposes.

For positive biasing, there is additional problem. If a high positive bias is applied to one terminus of an array, the array will achieve a potential distribution with respect to the space plasma so as to equalize the positive and negative currents to the array surface from the space plasma. Since the electron current density will be much greater than the ion current density, the positively biased terminus will float somewhat positive while the opposite terminus will float substantially negative (surface areas will balance current densities). The opposite terminus (negative with respect to the space plasma) will then be susceptible to arcing problems discussed above for negatively biased arrays. If the solar array makes up most of the surface area of the spacecraft and the opposite terminus of the array is connected to the frame of the spacecraft (i.e., grounded), then not only the opposite end of the array but also other parts of the spacecraft could become susceptible to arcing.

One way of preventing swinging the vehicle (or the other terminus of the array) negative when applying positive bias is to turn on an electron emitter (e.g., a tungsten filament). The outgoing electron current produced will balance the incoming electron current to the positively biased part to the array without the radical altering of vehicle potentials (with respect to the space plasma) cited in the previous paragraph. PASP Plus has incorporated the use (and non-use) of an electron-emitting filament in its high-voltage biasing operations.

The wide altering of vehicle potentials when biasing one array terminus highly positive occurs for large dielectric (i.e.,planar) arrays. For concentrators (mini-Cassegrainian or SLATS), the principal reception area is not the solar-cell surface but a metalized reflecting area (which may be covered by a thin protective dialectric coating). Hence, for concentrators there may or may not be any wide altering of vehicle potentials when a large positive voltage is applied (or generated) at one terminus of an array. There might be no need (or utility) in employing an electron emitter to prevent wide vehicle potential swings. Since PASP Plus has both planar and concentrator arrays, this difference in interaction effects can be appropriately investigated by the PASP Plus experiment.

EXPERIMENT STATUS

The Preliminary Design Review (PDR) for the PASP experiment was held in January 1986 and the Critical Design Review (CDR) in June 1987. Following the Jan 1988 annual review of PASP, the Geophysics Laboratory (after finalizing the breakup of the IMPS payload) made the decision to incorporate PASP-relevant sensors from IMPS. Work on the new PASP Plus experiment went forward at JPL with the pressure gauge, Langmuir probe, and TPM supplied as GFE (Government Furnished Equipment), as were the four array modules supplied by the Aero Propulsion and Power Laboratory. Since no firm spaceflight for PASP Plus has been identified, there was no reason or basis (interface information) to fabricate an actual PASP Plus flight unit. GL decided to conclude JPL's present effort with a brassboard demonstration model of PASP Plus with full working electrical configuration but not mechanical flight configuration. Completion of the fabrication and testing of the PASP Plus brassboard unit is expected in September 1989. The GFE items are considered to be flight hardware. Examples of fabricated circuit boards for the ASIS (array selection and instrumentation system) and DACS (data acquisition and control system) portions of PASP Plus are shown in Figures 5 and 6.



Figure 5. PASP Plus ASIS I-V Curve Subsystem Board.

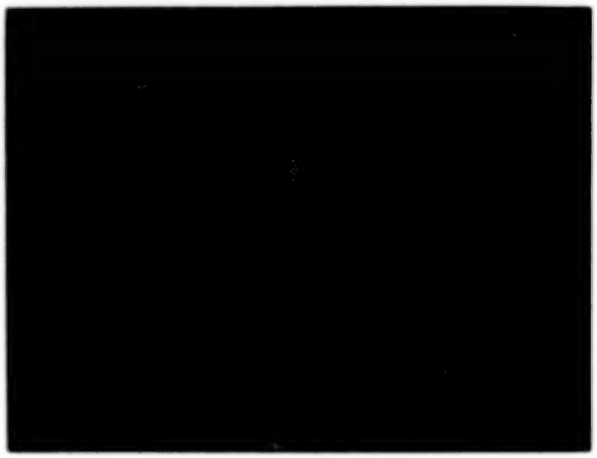


Figure 6. PASP Plus DACS Thermal Detector Controller Board.

EXPECTED RESULTS FROM PASP PLUS

Because of the complex characteristics of the space-plasma environment, the interaction effects on new-technology solar arrays cannot be calculated or convincingly simulated in ground tests. Results from a successful PASP Plus experiment will facilitate the fielding of new photovoltaic technologies through the validating and upgrading of computer models, development of CAE tools, and enhancement of ground-based simulation and testing.

We hope that arrangements can be made for a flight of the PASP Plus experiment in the reasonably near future (FY91-92). If PASP Plus achieves such a ride, we expect that the results will be thoroughly analyzed, both from an array performance viewpoint and an interaction phenomenology view-point. The performance of each of the planar and concentrator arrays will be measured at the various bias levels and under various space-plasma environment conditions (ram, wake or in-between orientation; passage through an auroral region if a high-inclination ride is obtained). Increases in radiated and power-line noise during high-voltage biasing will be measured. Short-term or long-term changes in array performance will be correleted with environmental condition changes determined by the diagnostic sensors. From all the data, we will try to ascertain cause-and-effect relationships.

Within the first year after a successful flight, the Geophysics Lab and the Aero Propulsion and Power Lab will conduct a series of workshops at which correleted PASP Plus data would be made available to the space-power and space-systems communities (in DoD and NASA). The workshops will be targeted to key topics such as concentrator performance, high-voltage operation, EMI-generation effects. Reports on these key topics resulting from these workshops will be directed towards upgrading relevant space-power design guidelines and test standards.

CONCLUSIONS

The PASP Plus complement of different types of solar arrays and diagnostic (interactions and environment measuring) sensors has been chosen to address the environmental sensitivities of the new technologies being developed. Failure to determine the extent of interaction problems by experiments such as PASP Plus could lead to serious flaws in future space-power subsystems.

REFERENCES

1. Snyder, D.B., "Discharges on a Negatively Biased Solar Cell Array in a Charged-Particle Environment," Spacecraft Environmental Interactions Technology - 1983, NASA CP-2359, AFGL-TR-85-0018, 1985, pp. 379-388.

2. Stevens, N.J., "Interactions Between Large Space Power Systems and Low-Earth Orbit Plasmas," Spacecraft Environmental Interactions Technology - 1983, NASA CP-2359, AFGL-TR-85-0018, 1985, pp. 263-276.

3. Grier, N.T., "Plasma Interaction Experiment II (PIXII): Laboratory and Flight Results," Spacecraft Environmental Interactions Technology - 1983, NASA CP-2359, AFGL-TR-85-0018, 1985, pp. 333-348.

4. Ferguson, D.C., "The Voltage Threshold for Arcing Solar Cells in LEO--Flight and Ground Test Results." AIAA 86-0362, AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, Jan. 6-8, 1986.

5. Stevens, N.J., "Summary of Pix-2 Flight Results over the First Orbit," AIAA-86-0360, AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, Jan. 6-9, 1986.

6. Jongeward, G.A., Katz, I., Mandell, M.J., and Parkes, D.E., "The Role of Unneutralized Surface Ions in Negative Potential Arcing", IEEE Trans. Nucl. Sci., Vol. NS-32, No. 2, Dec. 1985.

7. Parks, D.E., Jongeward, G.A., Katz, I., and Davis, V.A., "Threshold Determining Mechanisms for Discharge in High-Voltage Solar Arrays", AIAA-86-0364, AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, Jan. 6-9, 1986.

ORBITAL DEBRIS AND METEOROID EFFECTS ON SPACECRAFT SYSTEMS

E. Christiansen
NASA/Johnson Space Center

(Paper not provided by publication date.)

ADVANCED NETWORK ARCHITECTURES

Dr. J. Barhen
NASA/Jet Propulsion Laboratory

(Paper not provided by publication date.)

Sunspot Prediction Using Neural Networks

James Villarreal and Paul Baffes
Artificial Intelligence Section/FM72
Mission Planning and Analysis Division
National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas 77058

Introduction

The earliest systematic observance of sunspot activity is known to have been discovered by the Chinese in 1382 during the Ming Dynasty (1368 - 1644) when spots on the sun were noticed by looking at the sun through thick, forest fire smoke. Not until after the 18th century did sunspot levels become more than a source of wonderment and curiosity. Since 1834 reliable sunspot data has been collected by the National Oceanic and Atmospheric Administration (NOAA) and the U. S. Naval Observatory. Recently, considerable effort has been placed upon the study of the effects of sunspots on the ecosystem and the space environment. This chapter describes the efforts of the Artificial Intelligence Section of the Mission Planning and Analysis Division of the Johnson Space Center involving the prediction of sunspot activity using neural network technologies.

Sunspots

A sunspot is a dark region on the solar disk, indicative of a 2000°K cooler area than the normal photospheric temperature. On the average, sunspots are about 37,000 km in diameter (for comparison, recall that the earth's diameter is 12,740 km) with exceptionally large spots having a diameter of 245,000 km. Essentially, a sunspot is an eruption of a magnetic energy field extending several miles beyond the sun's surface with an accompanying sunspot of reversed polarity acting as a sink.

Therefore, sunspots are a basic measure of solar activity -- the more sunspots, the more active the Sun. Associated with these moments of high activity are increased occurrences of solar flares, which are bursts of electromagnetic energy. An eruption of a solar flare is accompanied by electromagnetic emissions in the microwave radio frequency range. The larger solar flares may emit relativistic charged particles and energetic protons.

The ability to predict sunspot activity plays an increasingly important role in both earth and space endeavors. Among the significant effects of sunspots are: x-ray emissions, energetic photons, ozone density fluctuations, solar wind variations, rainfall and temperature changes, and disturbances of the earth's geomagnetic fields. Such phenomena are important to NASA because of their adverse effect upon space environment. For example, x-ray emissions can disrupt radio communications by altering the electron density in the earth's ionosphere. Communication signals transmitted from radio stations are either refracted or reflected by the earth's atmosphere and returned to receiving stations. The electron density of the atmosphere, called "skin depth," determines the effects of the atmosphere on radio signals -- shorter wavelengths pass through the ionosphere whereas longer wavelengths are reflected. Consequently, any change in ionosphere electron density will disrupt radio communications and may even necessitate changes in transmission paths of navigation signals.

Other adverse effects by sunspots upon the space environment include the release or increased activity of energetic protons closely related to sunspot frequency. Such energetic protons can damage electronic components within satellites. Additionally, sunspots can cause fluctuations of the geomagnetic field resulting in heating of the earth's upper atmosphere. This causes increased drag on space structures and satellites, and complicates predictions of satellite orbits. In fact, increased atmospheric drag due to sunspot activity was the chief cause for the earlier than expected destruction of Skylab in the late 1970s.

Backpropagation Networks

Choosing an appropriate method for sunspot prediction requires a careful analysis of the desired output and the characteristics of the available data. The crux of the problem is to forecast future sunspot activity given a "window," or partial history, of past sunspot measurements. Because a *prediction* is being made, no traditional algorithm will suffice since future events will never exactly duplicate the past. In other words, a simple review of historical data will not work. One must be able to *generalize* from past measurements to have any hope of producing a hypothesis meaningful to an event which has yet to occur. Furthermore, a large amount of data has already been collected which can be brought to bear on the problem. It seems only logical to use as much of that data as possible to bolster the efficacy of the generated results.

In short, the solution method used must be able to digest the available data into patterns which have some significance to forecasting. One neural network paradigm in particular, called the "generalized delta rule" or "backpropagation," fulfills all of the requirements outlined above. Given large sets of input data, backpropagation networks can be used to categorize input patterns *never before presented* to the network. This categorization is not a simple lookup of past events but the result of generalization on the input data based upon a blending of its various features. To show how this can be accomplished, it is instructive to understand the general structure of a backpropagation network.

Background

As the name implies, artificial neural networks are based on concepts borrowed from biological nervous systems. Anatomical evidence of the nervous system indicates that single neurons are highly interconnected to other neurons with which they communicate through the release of variable amounts of neurotransmitters at the synapse. By modelling these properties in computer systems where interconnections are highly distributed and each element is treated as an individual parallel processor, interesting and useful properties have surfaced. Artificial neural networks have the unique property of being able to automatically extract and develop internal features from a given data set and to form generalities from those learned features.

Highlighting the mechanics of artificial neural networks may best be done by comparing the differences between artificial neural networks and the conventional computer system. Conventional computer systems generally consist of a centralized processing unit and an addressable memory. The central processing unit accesses locations of memory where information can be stored or retrieved. This structure is analogous to a postman who stuffs letters into mailboxes. Artificial neural networks, on the other hand, consist of numerous simple processing elements which are highly interconnected. It is in the *connections between* processing elements and *not* in the processing elements themselves where information in a neural network lies. Therefore, a network's memory is not stored in discrete locations as with a conventional computer. Instead, information is *distributed* throughout the entire network and is retrievable only through the interactions of its various processing elements. Unlike conventional computer systems where information is retrieved or fetched from memory, an artificial neural network can best be described as *evoking* its stored information.

Processing Elements

As mentioned earlier, a network is comprised of numerous, independent, highly interconnected processing elements. For backpropagation networks, each element can be characterized as having some *input* connections from other processing elements and some *output* connections to other elements. The basic operation of an element is to compute its *activation value* based upon its inputs and send that value to its output elements. Figure 1 shows a schematic of a processing element. Note that this element has j input connections coming from j input processing elements. Each connection has an associated value called a *weight*. The output of this processing element is fashioned to nonlinearly transform its summed, continuous-valued inputs by the sigmoid transformation shown by the two formulas in Figure 1. Understanding the details of this transformation is not essential here; the interested reader will find an excellent description of such details provided by Rummelhart et. al.[7]. For the purposes of this discussion it is important simply to note that a processing element's output is calculated solely from the influence of its incoming elements and connections.

When groups of processing elements are arranged in sequential layers, each layer interconnected with the subsequent layer, the result is a wave of activations propagated from the input processing elements, which have no incoming connections, to the output processing elements. The layers of elements between the inputs and outputs take on intermediate values which perform a mapping from the input representation to the output representation. It is from these intermediate or *hidden* elements that the backpropagation network draws its generalization properties. By forming transformations through such intermediate layers, a backpropagation network can arbitrarily categorize the features of its inputs. More importantly, since these categorizations are formed by summing the effects of the inputs, the result is a generalization over the input vector.

$$E_i = \Sigma w_{ij} p_j$$

$$p_i = P(E_i) = \frac{1}{1 + e^{-E_i}}$$

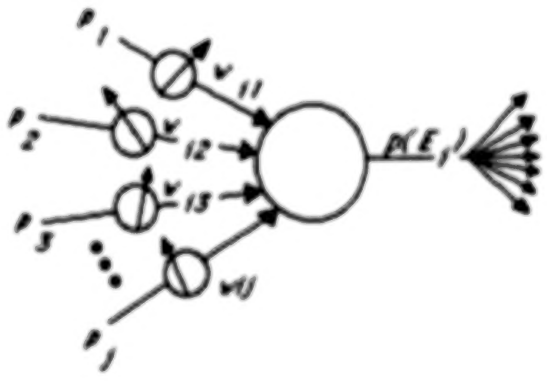


Figure 1: Processing element in a backpropagation network.

The Weights of a Backpropagation Network

The heart of the backpropagation algorithm lies in how the values of its interconnections, or weights, are updated. Initially, the weights in the network are set to some small random number to represent no association between processing elements. Upon being given a set of patterns representing pairs of input/output associations, the network enters what is called a *training* phase. During training, the weights are adjusted according to the learning technique developed by Rumelhart et. al. The training phase is modelled after a behavioristic approach which operates through reinforcement by negative feedback. That is, the network is given an input from some input/output pattern for which it generates an output by propagation. Any discrepancies found when comparing the network's output to the desired output constitute mistakes which are then used to alter the network characteristics.

According to Rumelhart's technique, every weight in the network is adjusted to minimize the total mean square errors between the response of the network, p_{pi}, and the desired outputs, t_{pi}, to a given input pattern. The indices p and i represent the pattern number and the index to a node respectively. The weights are adjusted according to:

$$\Delta w_{ij}^{(n+1)} = \alpha \Delta w_{ij}^{(n)} + \eta \delta_i^{(n+1)} P_j^{(n)}$$

where $\Delta w_{ij}^{(n)}$ is the error *gradient* of the weight from the jth processing element in layer n to the i th unit in the subsequent layer $(n + 1)$. The parameter α, performs a damping effect through the multi-dimensional error space by relying on the most recent weight adjustment to determine the present adjustment. The overall effect of this weight adjustment is to perform a gradient descent in the error space; however, note that true gradient descent implies infinitesimally small increments. Since such increments would be impractical, η is used to accelerate the learning process. Finally, the error signal, δ_i, is first determined for the output layer, N:

$$\delta_i^{(N)} = (t_i - p_i^{(N)}) P'(E_i^{(N)})$$

and then recursively back propagated through the higher layers:

$$\delta_i^{(n)} = \sum_j \delta_j^{(n+1)} w_{ji}^{(n)} P'(E_i^{(n)})$$

where $P'(E)$ is the first derivative of $P(E)$.

Again, the fine details of the above equations are left to the more thorough discussion provided by Rumelhart, though some important features should be emphasized. First, note that each weight is changed according to a gradient descent technique. This implies that the training process is meant to converge on some minima in the error space. The network is said to have *learned* if the error at this point is below the desired threshold set by the user at which point no further training is performed. Loosely speaking, this implies that the more weights present, the larger the error space and, in general, the larger the number of minima at which the network can be satisfied. The implication, then, is that using larger hidden layers which require more weights will help the network to converge. Unfortunately, added "convergence power" is not the only effect of increasing the numbers of hidden processing elements. Larger hidden layers adversely influence the generalization capabilities of a backpropagation network. In short, the network simply "memorizes" the training patterns. It is only through decreasing the number of hidden processing elements that backpropagation networks can be forced to generalize.

At present, tradeoffs such as these are a typical part of designing neural networks. Setting the number of hidden processing elements, as well as determining values for the constants α and η, is still somewhat of a "black art" best mastered through experience. What follows is our experience setting these parameters for a backpropagation network used to predict sunspot activity.

Sunspot prediction

Several key issues must be considered when designing with a backpropagation network. These issues may include data preprocessing, data format presentation to the network, the network architecture, and the tests for generality. Other concerns include the number of training patterns and training cycles required for successful generalization. Experience has shown that, if possible, neural networks are easier to analyze and manage when the data is processed before it is presented to a neural network. Common forms of processing include normalization of the data, separating the data or system into its constituent forms, and compacting the data into non-redundant formats. Figure 2 illustrates the sunspot data as supplied by the NOAA; monthly sunspot numbers lie along the ordinate and time ranges from years 1834 to 1984 along the abscissa. The mathematics which describe the properties of the backpropagation are not well understood. Throughout the discussion, it will become apparent that neural network designs are primarily empirical. Therefore, we will focus on the issues critical to the development of a successful and usable neural network.

Selecting the training set

An immediate observation is the 11 year cyclic period evident throughout the supplied data. It was this observation that led to the selection of 132 months as the input to the neural network. The neural network had a task which is difficult even by human pattern recognition standards -- to predict future solar activity based solely on historical observations. As is evident in the data, the neural network had to automatically categorize between the very high frequencies apparent on a month by month scale, the middle frequency ranges or the 11 year cycle, and an even closer examination reveals a much lower frequency suppressing or intensifying magnitudes of sunspot activity clusters. The physical underlying phenomena behind these solar characteristics are not well understood by solar experts nor will the neural network be able to explain them.

However, solar science phenomenas were neglected and the data was fashioned so that the neural network's only goal was to predict a future sunspot level. Therefore, the neural network was presented 11 years of data at the input and the output represented the associated future month for a particular input pattern.

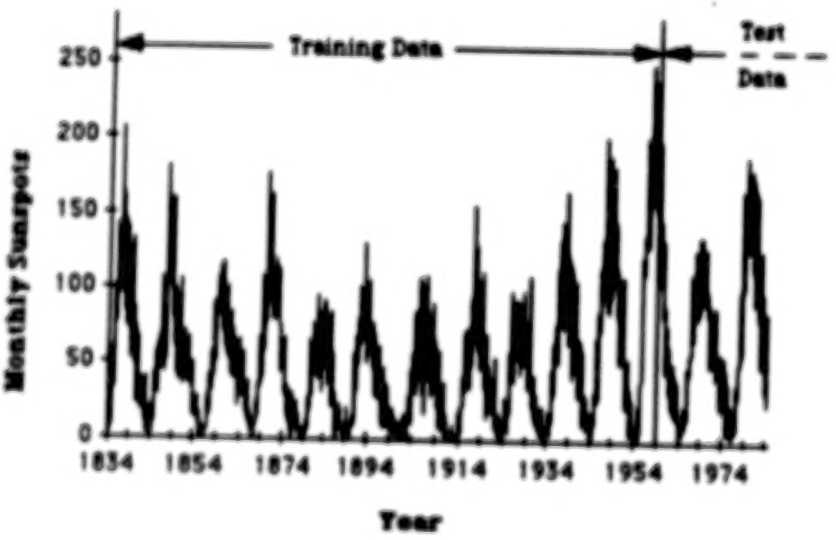


Figure 2: Monthly sunspot numbers from 1834 to 1984 as provided by the NOAA

All processing elements throughout the network used a non-linear activation function. When the upper and lower bounds of the sigmoidal transfer function are set at 1 and 0, respectively, close observation reveals that the most linear region of the processing element's output is exhibited in the range from 0.2 to 0.8. Therefore, the entire sunspot data was normalized between these two ranges; i.e., a value of 0.2 represented no sunspot sightings whereas a value of 0.8 represented a maximum of 254 sunspot sightings (maximum sunspot sightings observed for the provided epoch).

Careful consideration must be given when developing a system representation for a neural network. Sejnowski, in his development of NETtalk, a neural network which learned the relationships between the English language and phonetics, used the 1000 most commonly used words in the English language to train NETtalk. The belief was that a 1000 word set was rich with a sufficient range of English to phonetic translations to cover a large percentage of the rules necessary to read or pronounce a word. With NETtalk, there is the danger of selecting such a small set of words, say 50, that the network would undergeneralize or not pronounce untrained words correctly or to provide such a large training set, say 20,000 words, that the network would have difficulty making distinctions and not adjust its weights correctly. Again, no real specification or rule of thumb exists which can assist in selecting the appropriate training set. Until further advances in neural networks are made, empirical methods seem to be the only solution here. The sunspot prediction neural network was experimented with varying sets of resolution in the training patterns; the input windows were incremented by 1, 4, 8, 10, and 20 month steps. Increasingly better prediction performance was found with increasing step size, maximizing at 10 month steps, and decreasing prediction performance at 20 month steps.

Network Architecture

Another key factor in developing successful neural networks deals with the construction of the appropriate neural network architecture. An earlier project demonstrated that a neural network which generated speech signals from a phonetic type input could only converge with a two hidden layer architecture. Naturally, early efforts in sunspot prediction using neural networks were based on a two hidden layer network architecture. Experiments were conducted with several neural networks architectures which varied the numbers of processing elements in the hidden layers. Even though each neural network converged to an acceptable level, every network exhibited poor generalization capabilities. An interesting observation in the dual hidden layer sunspot prediction neural network architectures was that after satisfactory levels of convergence had been achieved, relatively little activation was present in any of the processing elements in the second hidden layer. This, however, was not the case for the speech generation neural networks. In fact, increased activity in the second hidden layer for the speech generation neural network serves as an indicator for successful generalization.

The next phase of this project experimented with single hidden layer neural networks. Again, the number of processing elements in the hidden layer were varied from 120, 80, 60, and 30 processing elements. Even though each neural network converged, only the neural network with 30 hidden processing elements displayed satisfactory levels of generalization.

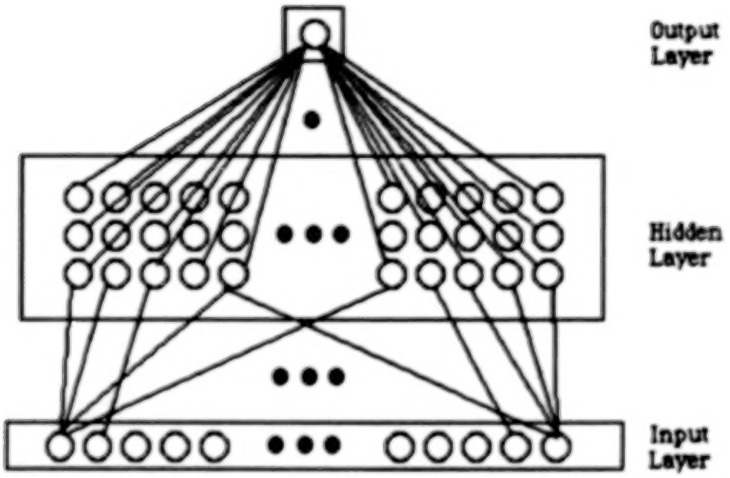


Figure 3: Neural network architecture used to predict a future month sunspot number given 132 (11 years) past sunspot numbers. The number of processing elements in the hidden layer were varied. All connections are not shown.

Testing the network

A neural network's worthiness is not measured by whether it can converge, but instead to how it can draw inferences or generalize to unforeseen stimuli. As noted earlier, each sunspot prediction neural network architecture experiment which varied the number of hidden layers, the number of processing elements in the hidden layers, and the complexity of the training data were all capable of converging. However, significant prediction capabilities were only discovered in a certain neural network architecture with a certain degree of training data complexity.

To test the generalization capabilities of the network, the network was trained with data from 1834 to 1959 and the state of the neural network's performance was tested against the remaining data which ranged from 1959 to 1984; i.e., the weights were not adjusted with the data from 1959 to 1984. Best generalization results were obtained when a 132 input, 30 hidden, and 1 output neural network was trained on a 10 month increment input pattern. Figure 4.0 is the RMS error of the output node with a 10 month step and 30 hidden processing elements. An interesting observation, which did not appear in any other error curve, is the crest found in the neighborhood of 500 passes. Whether the pattern in the error curve has any relevant significance to generalization is not known.

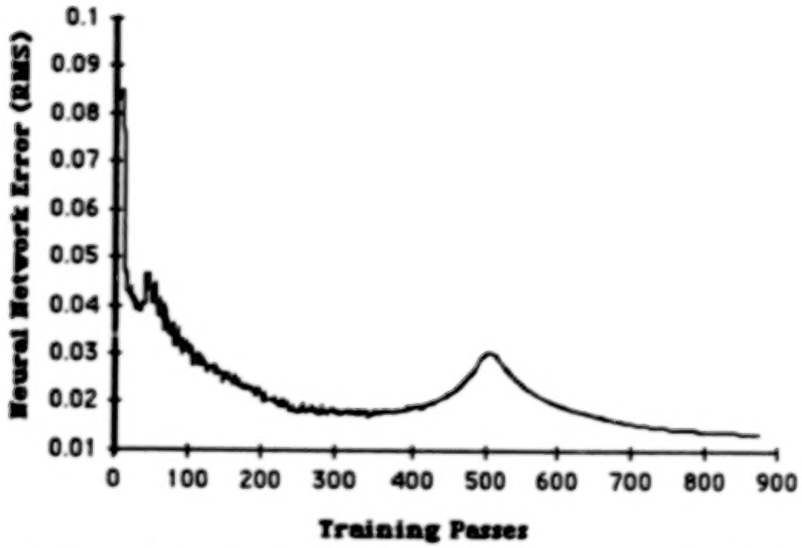


Figure 4: Root Mean Square (RMS) error for the output of a 132-30-1 neural network where the 11 year input window is incremented by 10 months as it traverses through the training data.

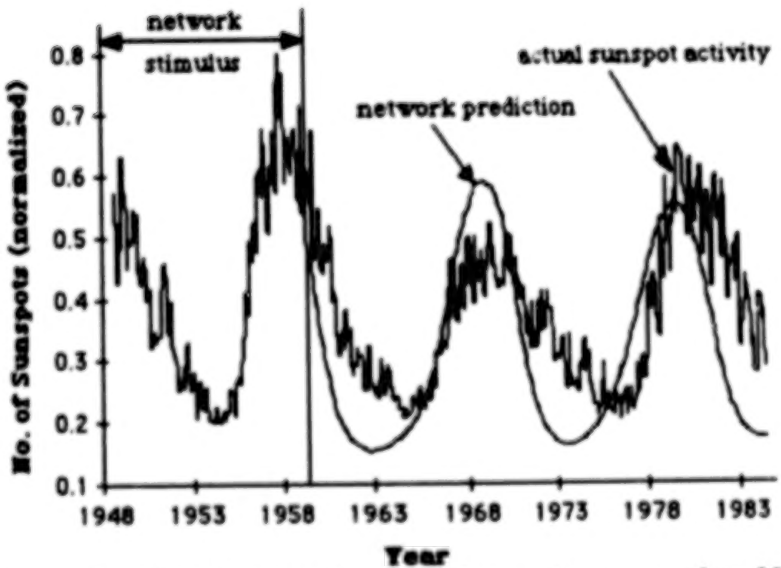


Figure 5a: Neural network's prediction performance after 25 passes (average error is 9.4% and RMS error is 11.3% for test segment).

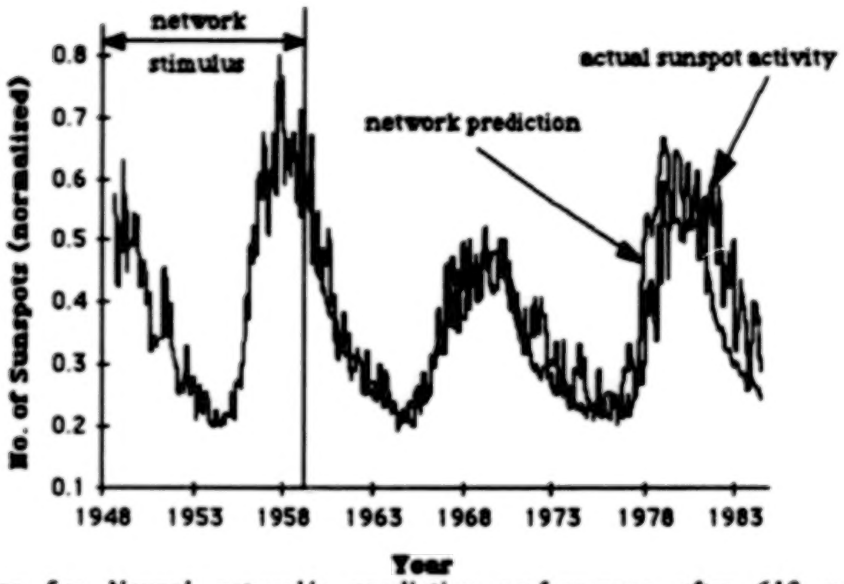


Figure 5c: Neural network's prediction performance after 610 passes (average error is 3.35% and RMS error is 4.58% for test segment).

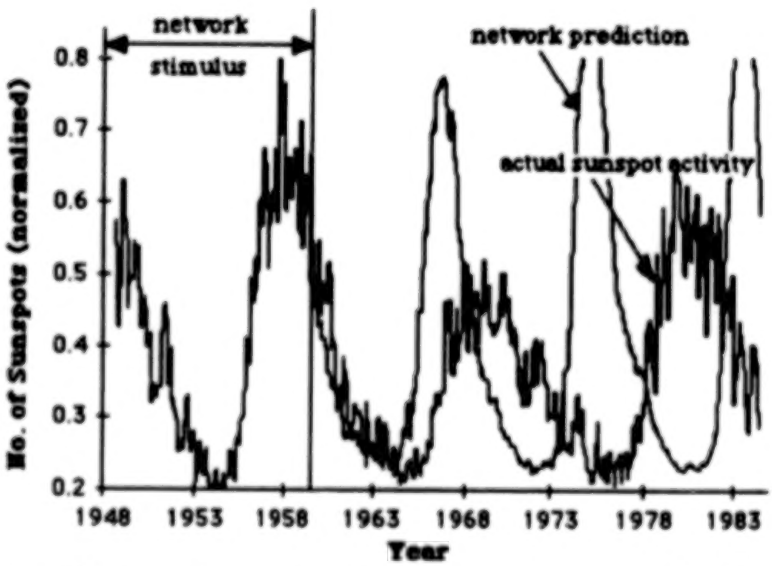


Figure 5b: Neural network's prediction performance after 260 passes (average error is 12.24% and RMS error is 18.16% for test segment).

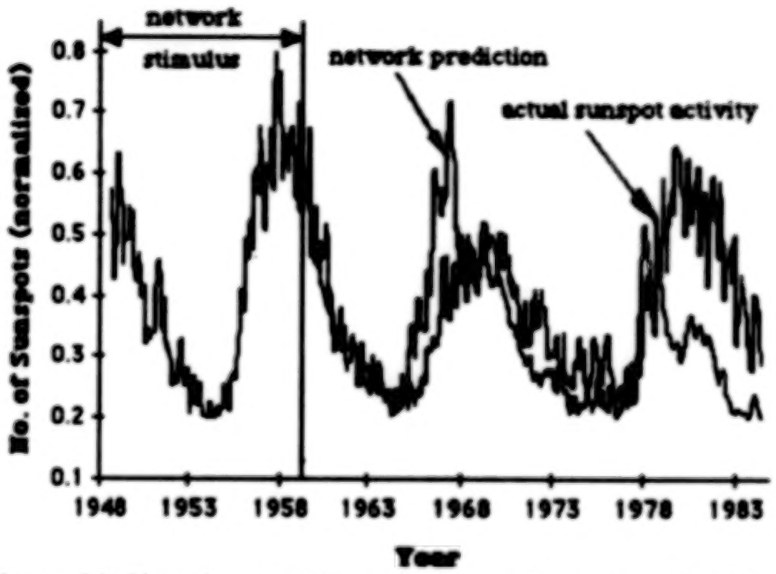


Figure 5d: Neural network's prediction performance after 810 passes (average error is 5.9% and RMS error is 8.6% for test segment).

Figure 5 illustrates several performance generalization states of the 132-30-1 neural network during the training process. The neural network's prediction capabilities were determined by suspending the "learning" process and then evoking the neural network with the last 132 months in the training set. The neural network's "prediction" or output was then appended to the tail end of the excitation source. This process was repeated for all the data points necessary to cover the remaining test portion of the actual sunspot number data.

The graphs show that, as expected, the neural network performed poorly in the early stages of training, peaked in the intermediate stages, and then returned to relatively poorer performances with continued training. For actual sunspot prediction uses, this neural network would be retrained with the original training data, while again monitoring its prediction performance against the test data. Having achieved a satisfactory level of confidence in its predictability, the neural network would be invoked with the actual test data and prompted to produce future "unknown" sunspot activity.

Conclusion

In summary, this work shows that neural networks are indeed a very useful tool for developing system models. Several key concerns have been pointed out which are necessary in developing useful neural network based systems.

References

[1] Bray, R. J., and Loughhead, R. E., [1964] "Sunspots", Dover Publications, Inc, New York, New York.

[2] Johnson, G. G., and Newman, S. R., [1980]"Solar Activity Prediction of Sunspot Numbers - Predicted Solar Radio Flux", JSC-16390, Houston, Texas.

[3] Herman, J. R., and Goldberg, R. A., [1977] "Sun, Weather, and Climate", Dover Publications, Inc., New York, New York.

[4] McNish, A. G. and Lincoln, J. V., [1949] "Prediction of Sunspot Numbers", Transactions, American Geophysical Union, Vol. 30, Number 5, pp. 673-685.

[5] Newman, S. R., [1980] "Solar Activity Prediction of Sunspot Numbers (Verification) - Predicted Solar Radio Flux - Predicted Geomagnetic Indices A_p and K_p", JSC-16762, Houston, Texas.

[6] Pepin, R. O., Eddy, J. A., and Merrill, R. B., [1979] "Proceedings of the Conference on The Ancient Sun - Fossil Record in the Earth, Moon, and Meteorites", Boulder, Colorado.

[7] Rumelhart, D. E., and McClelland, J. L., [1986] "Parallel Distributed Processing: Explorations in the Microstructure of Cognition", MIT Press, Cambridge, Massachusetts.

[8] Sawyer, C., Warwick, J. W., and Dennett, J. T., [1986] "Solar Flare Prediction", Colorado Associated University Press, Boulder, Colorado.

[9] Sejnowski, T. J. and Rosenberg, C. R., [1986] " NETtalk: A Parallel Network that Learns to Read Aloud", Johns Hopkins University, Technical Report JHU/EECS-86/01.

[10] Villarreal, J. A., [1988] "Artificial Neural Network Directions Within the Artificial Intelligence Section/ NASA", Proceedings of the ISA-88 International Conference and Exhibit, Houston, Texas.

[11] Vitinskii, Y. I., [1962] "Solar Activity Forecasting", Translated from Russian, U. S. Department of Commerce, Clearinghouse for Federal Scientific and Technical Information, Springfield, VA.

[12] White, H., [1988] "Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Predictions", Proceedings of the IEEE International Conference on Neural Networks - 1988, Institute of Electrical and Electronics Engineers, Inc., New York, New York, pp. II-451 - II-458.

[13] Wang, J. C. H., [1980] "A note on sunspot records from China", Proceedings of the Conference on The Ancient Sun - Fossil Record in the Earth, Moon, and Meteorites, Boulder, Colorado.

BEHAVIORAL NETWORKS AS A MODEL FOR INTELLIGENT AGENTS

Nancy E. Sliwa
MS 152D
NASA Langley Research Center
Hampton, Virginia 23665-5225

ABSTRACT

This paper describes on-going work at NASA Langley Research Center in the development and demonstration of a paradigm called behavioral networks as an architecture for intelligent agents.

This work focuses on the need to identify a methodology for smoothly integrating the characteristics of low-level robotic behavior, including actuation and sensing, with intelligent activities such as planning, scheduling, and learning. This work assumes that all these needs can be met within a single methodology, and attempts to formalize this methodology in a connectionist architecture called behavioral networks.

Behavioral networks are networks of task processes arranged in a task decomposition hierarchy. These processes are connected by both command/feedback data flow, and by the forward and reverse propagation of weights which measure the dynamic utility of actions and beliefs.

An experimental prototype of a behavioral network testbed is being developed in the Intelligent Systems Research Lab. This work is augmented by grants with Old Dominion University and the University of Maryland.

JUSTIFICATION

As NASA's mission repertoire continues to favor large, complex, long-duration missions, design and operations costs and manpower commitments could come to dominate NASA's budget and activities. This would limit NASA's ability to start new programs, hampering NASA's quest to continue expanding the frontiers of knowledge, understanding, and technology. Operational activities must be made less resource-demanding, more efficient. Increased assistance from computers and intelligent systems is one possible means of maintaining future flexibility in operational commitments.

NASA is currently on the threshold of operational deployment of its first-generation artificial intelligence systems. It appears that at the current state of the practice, the best payback to NASA is in the development of relatively small, single-purpose expert systems. These systems are appearing in launch processing, mission control[1], Shuttle-based experiments, and are baselined for elements of Space Station Freedom operation.

However, NASA's future mission plans call for elaborate, complex, and interconnected systems that integrate not only different functionality, but which span the multiple spectra of symbolic and numeric computation, human and robotic activity, and high and low speed and bandwidth requirements. The class of tasks to be performed by such systems involve handling perception, cognition, action, and reaction with smooth simultaneity. The ideal system would also modify its behavior appropriately based on feedback and its history of performance, and be relatively easy to develop. Unless these future requirements are addressed today, the capability will not be available tomorrow.

Myriad research projects exist that ably address specific components of these needs, such as planning, resource allocation, and learning. This isolationist approach assumes that, after all problems are "solved" independently, the solution techniques can be stirred together into a complete system. It can be argued that unless all requirements are

considered together, this resulting composite system will fail to integrate those requirements in a satisfactory way.

A fundamental assumption of this research is that there exists a single methodology that smoothly blends all requirements into a single architecture. This research is concerned with developing such a methodology that provides a "seamless fit" among the broad spectrum of activities and abilities of an intelligent system, including planning, scheduling, resource allocation, execution control, perception, and learning. Systems developed under such a methodology would provide NASA with the type of intelligent systems required for future missions. This methodology should allow modules of different functionality to be developed using similar techniques and to work together smoothly. For example, consider a lunar outpost for LOX production. A power allocation system, a crew activity planning system, and the control systems of surface robots could potentially all be developed in the same framework, and would be able to act and interact intelligently.

This paper provides a general motivation and description of the behavioral network concept, and discusses some issues associated with this approach. Current work is summarized.

GENERAL MOTIVATION

There seem to be two general classes of approaches to intelligent system research. One is the development of a "bag of tricks," an accumulation of techniques that are applied as suitable to a particular problem type. This approach has worked well for the area of computer programming, and is typical of young, immature areas of technology. The second approach is the development of a general theory and methodology applicable across most if not all of the problems in the field. This approach is usually more successful with a mature, well-understood technology. However, it is important, even in young technology areas, to continually examine successful "tricks" and attempt to formulate unifying approaches. This research project falls in this latter class.

The original work in the Intelligent Systems Research Lab in intelligent task decomposition and control focused on hierarchical levels of activity. A system that connected a blocksworld procedural planner to a jointed manipulator with an end effector and simple sensors was developed[2].

Experience with this development revealed several desirable attributes of a methodology for intelligent agent development:

1. A methodology must be able to connect symbolic and numeric programming approaches.

2. A methodology must be able to connect slow processes and fast processes.

3. The environment and the goals of an intelligent agent change dynamically, both from its own actions and from changes in the external environment. A methodology must unite goal-driven planning and reactive planning in a cooperative way.

4. A methodology must be able to blend control from both the intelligent agent itself and a human operator.

5 Sensor-closed control loops are very effective. A methodology should incorporate them.

6. The concept of hierarchical levels is relative; a function can be "higher level" or "lower level" than another, but architectures which defines precise levels are forcing arbitrary cuts in a continuum for the sake of convenience.

7. The same is true of the concept of heterarchy. A function has more or less interaction with other functions. Heterarchical architectures make arbitrary cuts in this continuum for convenience.

8. Actuators and sensors can be treated isomorphically. Actuators have a sensory component (proprioception), and sensors have an actuation component (positioning and activation).

9. Most robotic system development efforts never consider a general solution to the resource allocation problem. Most such systems are very resource constrained, and use customized solutions to the problems of redundant resources or resource failure. A methodology should provide a resource allocation technique that handles these problems and provides maximum parallelization of activities though appropriate resource allocation.

DESCRIPTION OF BEHAVIORAL NETWORKS

Based on these observations, the concept of behavioral networks was developed[3]. Behavioral nets represent a hybrid among classical control techniques, artificial intelligence planning techniques, and connectionist approaches.

Fundamentally, a behavioral network can be thought of as an acyclic directed graph whose nodes represent specific functions, or behaviors, of an intelligent system, with two-way links which propagate information including functional parameters and weights. The net flows from top to bottom in the task decomposition sense. That is, a node is linked downward, to "children" nodes, if the accomplishment of the child node's function is required to accomplish the original node's function. Put another way, a node accomplishes its goal or function by instantiating subgoals in the form of children behavior nodes.

Functions

Each node is built from a "template" (figure 1), which can be represented as a classic feedback control loop, receiving input as to the desired state X_d from a parent node, and receiving feedback Y concerning the current state or setting from children nodes. In the general case, the node would be required to compute the current state X_C from Y, i.e. $X_C = f(Y)$, where Y is a vector of the feedback signals. The node function then computes the required command parameter vector C to the children nodes that minimize the difference between the desired and current states, (i.e.: $X_d - X_C \rightarrow 0$), passes C to the children nodes, and passes X_C upward to its parent nodes in turn. In classical planning research, this equates to the selection and instantiation of an operator schema to minimize the "distance" between current and desired states via means-ends analysis.

This functional aspect of behavioral nets is similar to the concepts of structured programming approaches in both intelligent planning research and in control theory. However, behavioral nets provide a continuous flow of control in an isomorphic structure from potentially high-level, symbolic behavior, to low-level numeric control functions, providing a way to smoothly integrate goal-oriented behavior and reactivity. In addition, given the



Figure 1. - Behavioral Network Node Template

command and feedback components of the node template, each behavior has some degree of both motor and sensory functionality, providing a way to isomorphically represent actuation and sensory behaviors.

Weights

Another aspect of behavioral nets concerns the propagation of "weights" or potentiation level/threshold measures. This gives behavioral nets their connectionist flavor. Commands (goals) from parent to child are weighted according to utility measures, including the probability of success and the need or urgency or the action. Feedback from child to parent is also weighted according to similar parameters from a reactive or sensory point of view.

These weights are combined within each node, at each execution cycle, and form updated weights for use in the next cycle. Thus weights are propagated both "upward" and "downward" in the network, as are the command parameters and feedback. A threshold switching function requires each node's weight to exceed an established threshold before the node can "fire." This threshold varies with the utility cost of an action.

In general, a node will have more than one parent, and

multiple children (figures 2 and 3). The choice of which child to activate when, and with what parameters, is determined by the combinations of weights at each cycle of the network. This provides a method for choosing among competing subgoals, and for sequencing subgoals.

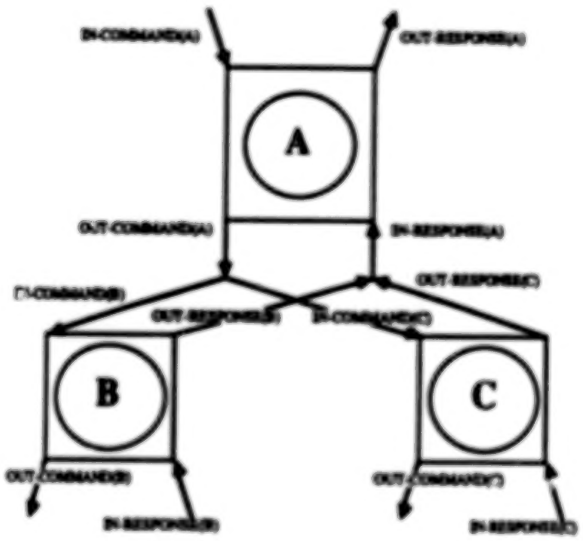


Figure 2. - Parent Node with Multiple Children

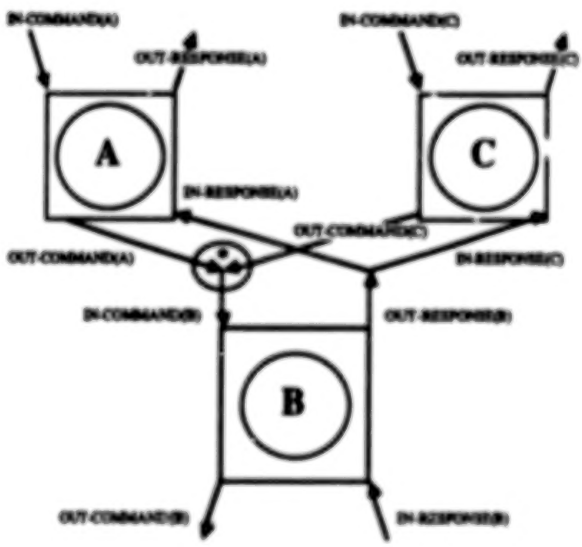


Figure 3. - Child Node with Multiple Parents

The weighting scheme represents a real-valued logic ranging between -1 and 1, with 1 representing total belief/desire, -1 representing total disbelief/avoidance, and 0 representing ignorance/indifference.

The weights serve several purposes. They determine when a behavior has been activated, and arbitrate among goal and resource conflicts. They provide a way to blend control from multiple sources, including from a human operator. They also alter the network over time according to the feedback, either increasing or decreasing the probability that a behavior is activated or that it successfully competes with other behaviors.

Parallelism

Parallel activity is an inherent part of behavioral networks. In the behavior net theory, each node is assumed to be continuously active if its threshold is exceeded, and is continuously checking its inputs and weight status. Thus the network is continuously adapting to new goals, and to changes in the environment. This also provides a technique for dealing with processes of varying speeds. Behaviors work at their own speeds asynchronously, using the most recently available command and feedback data from their parents and children.

True parallelism can of course only be achieved in a multi-processor environment. The ability of a behavioral network-based system to maintain its operational integrity in a time-shared environment would require careful design and static resource allocation.

RESEARCH ISSUES

Behavioral approaches are being investigated in other research groups. Brooks' work[4] in subsumption architectures is behavior-oriented, but does not attempt to establish a methodology. Brooks' goal is to build systems which "do the right thing;" the resulting architectures have highly convoluted wiring and logic, are not easily duplicated or understood, and are not extensible. Our research in behavioral networks, on the other hand, attempts to duplicate the functionality of Brooks' systems within a structured, easily developed, and extensible architecture.

Behavioral approaches have also been used quasi-operationally. The Hughes Corporation used a behavioral approach in the Autonomous Land Vehicle work for DARPA[5]. This system chose optimal path components from weighted alternatives. That approach is representative of most behaviorist research to date: techniques for choosing among alternatives, or disjunction. In addition to providing a structured approach to behavioral system development, this

research is also attempting to expand the abilities of such systems to handle conjunctive activities, both sequentially and in parallel. Several pertinent issues relating to these extensions are discussed below.

Conjunctive Behaviors

To discuss the issues involved with sequential behaviors, consider three activities, A, B, and C. Most connectionist systems attempt to accumulate evidence in favor of a choice among the alternatives: A or B or C. Most tasks, however, involve being able to sequence behavior: A and then B and then C. For example, to pour a glass of soda, get a glass, and then open the can of soda, and then pour the soda in the glass. If multiple resources are available, some tasks can be done in parallel: e.g., A while B, and then C. For example, if two people are available, one can get the glass while the other opens the can of soda. Both of these activities must be completed, however, before the soda can be poured into the glass. A and B are preconditions of C.

Behavioral network theory is attempting to determine a method for structuring the network and propagating weights that allow maximum parallelism while appropriately sequencing activities, as well as choosing among alternatives. Sequences and alternatives should emerge naturally as a result of the dynamic variation of the weights within the network, instead of being per force programmed into the code procedurally.

Goal Maintenance/Achievement

Goals and subgoals in an intelligent system are generally classed as goals of achievement or goals of maintenance. Goals of achievement are those that must be achieved at some point, but to which the system is indifferent thereafter. Goals of maintenance must be achieved and maintained for a period of time to establish the preconditions of later actions. Using our example above, the glass must continue to be available and the can must continue to be open until the action of pouring the soda into the glass is complete. Thus A and B must be maintained until C is achieved. The current research is working to establish a means of making this distinction intrinsically within the network. Precondition goals of maintenance are inherently resources that are required for subsequent actions. Weights propagated through the network should be able to maintain this logical resource availability, just as it obtains and maintains physical resources for tasks.

Network Structuring

The approach to decomposing a problem into subtasks has heretofore been very ad hoc. Many decompositions of one problem are possible, and no heuristics exist to rate one decomposition against another. Therefore, an effort is being made to formalize the decomposition task itself, in an attempt to optimize and eventually automate the decomposition process.

Lattice structures have long been known to offer a decomposition of partially ordered sets which exhibit algebraic structure. Dr. David Livingston at Old Dominion University developed a method to generate task decompositions by constructing the lattice of substitution property (SP) partitions on a state machine model of the task. Given the SP lattice, a group of partitions that will yield a "good" decomposition are selected[6]. Given the decomposition network resulting from this process, fourth-order constraint satisfaction networks have been able to find a path from the initial state to the goal state, thus generating a plan to perform the task.

This decomposition approach provided a method for finding decompositions, but has proved to be computationally intractable, and still reliant to some degree on heuristics. However, initial work by Dr. Livingston indicates that self-organizing networks could be used to find good, though non-optimal, decompositions in constant time. Continuing work under this grant is investigating more fully this approach, and integrating the task decomposition and planning techniques within the self-organizing network paradigm.

Connectionist Planning

A grant with Dr. James Reggia of the University of Maryland is currently investigating the application of connectionist competitive activation techniques to planning, scheduling, and resource allocation. Work to date has concentrated on searching the literature for connectionist approaches to similar problems, and on executing different problem solutions on the University of Maryland MIRRORS/II connectionist simulator. Three problem

domains and functions have been successfully developed: satellite camera resource cooperation to maximize target coverage[7], Voyager resource sequencing during planetary flybys, and fault interpretation and recovery for satellites[8]. These experiments demonstrate the ability of connectionist models to handle different function types, but more stringent problem models are necessary to determine their usefulness for complex problems.

Prototype System Development

A prototype implementation of a behavioral net system is being developed on a Symbolics 3620 in Lisp. This prototype, called Behavioral Network Functionally Integrated Testbed (BeNeFIT), will provide a means of testing and analyzing the performance of behavioral networks on a range of problems. This prototype will be used primarily to demonstrate the mechanics of the behavioral network structure, and to investigate various methods for propagating weights thoughout the network.

CONCLUSION

A methodology is in development to provide a structured and extensible approach to the design and development of behavior-driven intelligent agents. Behavioral networks are task decomposition networks which propagate commands, data, and feedback in a structured programming sense, and which propagate weights in a connectionist sense. Key research issues hinge on the ability of the network to represent the task in a logical way, to combine sequential, parallel, and alternative behavior in a single structure, and to handle the distinctions between subgoal achievement and maintenance. Supporting work at the University of Maryland and Old Dominion University is in progress. A prototype testbed implementation is being developed for NASA Langley Research Center for demonstration and research purposes.

REFERENCES

1. Muratore, John, "Application of Artificial Intelligence to Space Shuttle Mission Control," AAAI Conference on Innovative Applications of Artificial Intelligence, Stanford, California, March 1989.

2. Orlando, Nancy, "A System for Intelligent Teleoperation Research," AIAA Computers in Aerospace IV Conference, Hartford, Connecticut, October 1983.

3. Sliwa, Nancy and Donald Soloway, "A Lattice Controller for Telerobotic Systems," 1987 American Controls Conference, Minneapolis, Minnesota, June 1987.

4. Brooks, Rodney, "A Robust Layered Control System for a Mobile Robot," IEEE Journal of Robotics and Automation, RA-2(1), April 1986.

5. Daily, M., Harris, J., et al., "Autonomous Cross-Country Navigation with the ALV," Proceedings of the DARPA Knowledge-Based Planning Workshop, Austin, Texas, December 1987.

6. Serpen, G. and D. Livingston, "A High-Order Boltzmann Machine Configuration for Transfer Sequence Searches in Decomposed State Machines," presented at the International Joint Conference on Neural Networks, Washington, D.C., June 1989.

7. Bourret, P., Goodall, S., and M. Samuelides, "Optimal Scheduling by Competitive Activation: Application to the Satellite Antennas Scheduling Problem," Proceedings of the International Joint Conference on Neural Networks, Washington, D.C., June 1989.

8. Bourret, P. and J. Reggia, "A Fast Interactive Satellite Failure Diagnosis Procedure," Proceedings of the Goddard Conference on Space Applications of Artificial Intelligence, Greenbelt, Maryland, 1989.

AUTOCLASS II

P. Cheeseman
NASA/Ames Research Center

(Paper not provided by publication date.)

OMS FDIR - Initial Prototyping

Eric W. Taylor and Matthew A. Hanson
Ford Aerospace Corporation

ABSTRACT

The Space Station Freedom Program (SSFP) Operations Management System (OMS) will automate major management functions which coordinate the operations of onboard systems, elements and payloads. The objectives of OMS are to improve safety, reliability and productivity while reducing maintenance and operations cost. This will be accomplished by using advanced automation techniques to automate much of the activity currently performed by the flight crew and ground personnel. OMS requirements have been organized into five task groups: 1) Planning, Execution and Replanning, 2) Data Gathering, Preprocessing and Storage, 3) Testing and Training, 4) Resource Management, and 5) Caution & Warning and Fault Management for onboard subsystems. The scope of this prototyping effort falls within the Fault Management requirements group. The prototyping will be performed in two phases. Phase 1 is the development of an onboard communications network FDIR system. Phase 2 will incorporate global FDIR for onboard systems. Research into the applicability of expert systems, object-oriented programming, fuzzy sets, neural networks and other advanced techniques will be conducted. This paper discusses the goals and technical approach for this new SSFP research project.

INTRODUCTION

The Space Station Freedom Program (SSFP) Operations Management System (OMS) is a set of functions which includes application software and manual interactions with the Freedom Station either from onboard or on the ground. OMS requirements have been organized into five task groups: 1) Planning, Execution and Replanning, 2) Data Gathering, Preprocessing and Storage, 3) Testing and Training, 4) Resource Management, and 5) Caution & Warning and Fault Management for onboard subsystems. The scope of this prototyping effort falls within the Caution & Warning (C&W) and Fault Management requirements group. The purpose is to address the global fault detection, isolation, and reconfiguration (FDIR) requirements for OMS automation within the Space Station Freedom program.

Initial prototyping will concentrate on Network Management FDIR, a user interface, and a modular system architecture which incorporates advanced automation (i.e. expert systems, neural networks, etc). Subsequent prototyping will integrate the initial prototype with other prototype activities which are currently being developed in the SSFP End-to-end Test Capability (ETC) testbeds.

SSFP OMS BACKGROUND

The Command and Control Architecture for Integrated Operations Control, as defined in [PDDR88], specifies a hierarchical command structure with multiple tiers. Two levels are defined and are called Tier I and Tier II. "Tier I shall be the source of real-time and near real-time command and control to Tier II and shall provide top-level management and station-wide integration." "Tier II shall receive and execute commands from Tier I and provide the required information to

it."[PDDR88]

The OMS application software consists of an onboard portion which is designated as the Operations Management Application (OMA); and a ground resident portion which is designated as the Operations Management Ground Application (OMGA) [PDDR88]. The OMA is the onboard portion of the Tier I Command and Control Architecture. As such, the OMA interfaces with and provides top level command and control to the Tier II Distributed System Executives, Element Managers, and Payload Managers for integrated real-time and near real-time operations. Tier II, in turn, supports the OMA by providing management of their sub-components and by providing health, status, and other information, as required, to the OMA.

The OMGA is the ground application software portion of the Tier I Integrated Operations Control of the OMS. The OMGA is the only command access from the ground to the OMA. It performs unique functions in support of ground activities for operations, complements OMA functions in support of operations, and monitors OMA Activities [PDDR88].

In general, the Tier II managers are responsible for their own FDIR. The OMA performs station-wide FDIR when more than one Tier II manager is affected. The OMGA augments the OMA's actions for station-wide FDIR. Once the C&W function annunciates a failure or anomaly, the FDIR function identifies the source of the failure and the necessary actions to be taken. [PDDR88]

TECHNICAL APPROACH

Initially, prototype functionality will be targeted for onboard communication network FDIR. This portion of the prototype represents a Tier II Executive. It will communicate with network management entities within the internet to provide a central point of network management control. The Tier I OMA FDIR will communicate with this Tier II Network Executive providing the integrated onboard network FDIR aspect of Station-Wide Fault Management (SWFM).

The ISO model for Open Systems Communications defines services and protocols which provide for the exchange of network management information between Network Managers and Agents. MAP/TOP is a specification of the ISO standards which define this communication between Network Managers. Currently, these functions are not completely specified or available but it is assumed that the ISO standards (and possibly the MAP/TOP specification) will be used on the SSFP communications networks. We intend to use these types of functions to exchange information between the Tier II Network Executive and other network managers within the internet to perform communications network fault management.

When this is on solid ground, the second phase of this prototyping effort will develop a Tier I Manager which initially communicates with the above Tier II Network Executive. Subsequently, the prototype will include Station-Wide Fault Management, communicating with subsystem prototypes being developed on related SSFP prototyping projects (e.g. EPS, ECLSS, GN&C, OMS Short Term Planning, etc.).

Two prototype systems are currently envisioned: an OMA system residing in the JSC DMS test bed, and an OMGA system residing in the SSCC test bed. Towards the end of this project, end-to-end demonstration scenarios are planned within this portion of the End-to-end Test Capability (ETC) environment.

ADVANCED AUTOMATION IN OMS FDIR

The goal of this prototype effort is to exhibit the utility of knowledge based systems and other advanced automation techniques in Space Station Freedom program FDIR. For example, production systems may be employed during fault isolation and recovery where expert knowledge of system-wide faults can be utilized. Object Oriented Programming (OOP) and Frames may be used to create and maintain a model of the network (both at Tier I and Tier II levels) which will aid in the fault isolation phase. Neural Networks and Fuzzy Logic applications will be explored as the project progresses.

A knowledge based system will contain the knowledge required to perform fault isolation and recovery given fault and status messages from systems under its influence. These messages include Caution and Warning and will be collected by the FDIR system either solicited or unsolicited. The system will proceed to determine the "root" fault from among "side-effect" faults from these messages, reaching a conclusion from the available information or initiating tests to acquire additional information.

Expert knowledge will be used to determine which faults to pursue and which tests to initiate. Which of the faults (if multiple), is most severe, thus requiring immediate attention? Which test will most likely return the most informative data at the least expense?

Faults which appear, and subsequently clear will be tracked and chronologically logged, building a database of problem areas. This information can then used during later FDIR functions. All information required to make a correct diagnosis will not always be available, so the system will be able to deduce from incomplete data.

The human OMS Operator will not be subjected to large volumes of "raw" alarm data. Recovery procedures dealing with faults that have been determined as not requiring operator intervention will be performed automatically. Many recovery procedures will require the assistance of flight crew or ground support personnel, such as the off-line test of a piece of equipment or the swap or repair of a board or component. Other recovery procedures could recommend contacting the responsible vendor. After a recovery procedure has been followed, the FDIR system will initiate tests on the affected system/component before the fault instance is "closed".

SOFTWARE ENVIRONMENT

There are three major aspects to the prototype's software development: 1) the knowledge-based modules, 2) the user interface, and 3) the underlying system code. "Underlying system code" refers to the code that integrates the user interface, knowledge-based modules, operating system, network functions, etc. In short, it includes all code except the user interface and the knowledge-based modules.

Some basic assumptions have been made about the software environment of this effort as follows:

- o Unix (or Posix) operating system
- o X-windows application/user interface.
- o ISO network standards

While some non-Unix options are considered for early prototyping proof-of-concept, Unix is the ultimate target operating system environment.

KNOWLEDGE-BASED SYSTEM

Two production systems were considered for this prototyping effort: CLIPS and Ops83. CLIPS was developed by JSC MPAD, and Ops83 was developed by Charles Forgy's Production Systems Technologies. Ops83 has proven to be the fastest production system benchmarked, and it contains a full-featured procedural language for top-level production system control. In addition, Ops83 offers full control of the recognize-act cycle and readily interfaces to C, Ada and Fortran language modules. Procurement of Ops83 has been initiated for this project.

USER INTERFACE

The Transportable Application Environment (TAE) is a user interface development

tool which runs under Unix and X-Windows. It provides the user interface developer with an interactive tool to create windowing hierarchies with associated menus, buttons, icons, slide bars, animated graphics, etc. The TAE system generates "C" code which references X-Window's Xlib. The developer then incorporates his application code into the code generated by TAE. This system will be used to efficiently generate the FDIR system user interface(s).

UNDERLYING CODE

The "C" programming language will be used for initial prototyping. The C++ Object Oriented Programming Language will be used and further applicability will be explored. As the project progresses, the Ada language will be used as it is the required language of the Space Station Freedom Program.

HARDWARE ENVIRONMENT

A secondary goal of this project is to demonstrate OMGA and OMA FDIR functionality in the ETC environment. This implies hosting in both the SSCC test bed (for OMGA) and the DMS test bed (for OMA). Initial prototyping will be carried out in the SSCC test bed for both systems. Selected software methodologies should make any ports to the DMS test bed insignificant efforts. Sun (Unix) workstations are available in the SSCC test bed for this project, and all supportive software (TAE Plus, C++, and Ops83) has been (or is being) acquired.

SUMMARY

An overview of our initial approach to this prototyping effort was given. The first area for prototyping is advanced automation in communications network management. This will be approached assuming the OSI Reference Model will be employed on Space Station Freedom communication networks. The second area of prototyping will be to integrate the network management prototype into the OMS testbed environment. Some background on the OMS environment was given. This second effort will require extensive research into the current testbed configuration and capabilities. In addition, open lines of communication between ourselves and the testbed players will need to be more fully established.

REFERENCES

[X3T987] ASC Technical Commitee, "FDDI Station Management (SMT)", Draft American National Standard X3T9/85-X3T9.5/84-49 REV-2.1 APR 87

[BAYE88] Bayer, Steven E., Harris, Richard A., et al, "A Review of Space Station Freedom Program Capabilities for the Development and Application of Advanced Automation" MTR-88D00059, MITRE Corp. for NASA/JSC, Houston, TX, DEC 88.

[BERG89] Berger, Eugene L., "Concept of Operations for Network Management of Space Station Freedom Onboard Networks" WP-88D0041, MITRE Corp. for NASA/JSC, Houston, TX, MAR 89.

[HUNT89] Huntington, Jill A., "OSI-based net management: Is It Too Early, or Too Late?", DATA COMMUNICATIONS, McGraw-Hill, Vol. 18, Number 3, MAR 89, pp. 111-129.

[PIET88] Pietras, J., McCoy, E., et, "Concept of Operations for Space Station Information System (SSIS) Network Integration Management, Part 1: Architecture and Functions", WP-88W00268, MITRE Corp. for NASA/JSC, Houston, TX, OCT 88.

[NOLE88] Noles, James, McCoy, Emily, et al, "A Network Services Reference

Configuration for the Space Station Information System", WP-88W00139, MITRE Corp for NASA/JSC, Houston, TX, OCT 88.

[CRON88] Cronk, Robert N., Callahan, Paul H., et al, "Rule-Based Expert Systems for Network Management and Operations: An Introduction" IEEE Network, IEEE Publishing Services, SEP 88, pp. 7-21.

[SUTT88] Sutter, Mark T., Zeldin, Paul E., "Designing Expert Systems for Real-Time Diagnosis of Self-Correcting Networks" IEEE Network, IEEE Publishing Services, SEP 88, pp. 43-51.

[MAPT87] MAP/TOP User's Group, "Technical and Office Protocol Specifications" Society of Manufacturing Engineers, Version 3.0, Draft, Implementation Release, MAY 87.

[PDDR88] Space Station Projects Office, "Space Station Projects Description and Requirements Document, Vol. 3: Projects Design and Development Requirements," JSC31000 Rev. D, pp. 159-194.

[ECKL89] Eckelkamp, Rick, et. al., "OMS Evolution Plan," Johnson Space Center, JAN 1989

[HCIG88] "Space Station Freedom Program Human-Computer Interface Guide," Version 2.1, DEC 1988

The Application of NASREM to Remote Robot Control

Michael W. Walker, Joe Dionise, Al Dobryden

Artificial Intelligence Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, Michigan 48109

Abstract

This paper describes the implementation of a remote robot controller, wherein the distance to the remote robot causes significant communication time delays. The NASREM telrobot control architecture is used as a basis for the implementation of the system. Levels 1 through 4 of the hierarchy were implemented. The solution to the problems encountered during the implementation and those which are unique to remote robot control are described.

1 Introduction

Remote robot control is becoming an increasingly important discipline in robotics. Undersea exploration, mining, and salvaging operations, and the inspection and maintenance of nuclear power plants are examples of terrestrial based applications. Space applications include the construction of large space structures, the in situ maintenance and repair of satellites, and the exploration of the planets and their moons. This relatively new area of robotics is characterized by large distances between the operator and the remote robots.

A testbed has been built for identifying and finding solutions to problems unique to this type of system. The operator station is located at Grumman Aerospace Corporation in Bethpage, New York. The remote robots are located at the University of Michigan, Ann Arbor, Michigan. Thus, the system is truly a remote robot control system.

This paper presents the design of the testbed. Although, the testbed has just been completed, some problems which have already been identified will also be discussed. We begin with a description of the control architecture. This gives a framework from which to describe various components in the system and to specify their modes and methods of interaction. The following sections present some of the details of the construction of the system. The final section concludes the paper.

2 Control Architecture

We have adopted the NASREM architecture for telerobot control [1]. The reference document defines the functional requirements and flight level specifications of the control system for the NASA Space Station IOC Flight Telerobot Servicer. We have used this document as a guideline for the development of the control system architecture. However, the physical remoteness of the robots begin controlled requires some extensions to this architecture. The problem basically stems from the communication delays between the local site where the operator is stationed and the remote site where the robots are located. This section presents a brief description of the NASREM architecture and the required extensions.

2.1 The NASREM Architecture

The NASREM architecture is a six level hierarchical control system as shown in figure 1. The outputs of each level are the inputs to the next lower level in the hierarchy. The inputs at each level of the hierarchy are:

1. Level 6 - Operations Control Level
   Inputs are commands to schedule the servicing of satellites.
2. Level 5 - Service Bay Control Level
   Inputs are commands to a service bay manager to perform operations on specific spacecraft.
3. Level 4 - Object/Task Level
   Inputs are commands to perform a task on an object in order to achieve a desired relationship of that object relative to other objects in the world.
4. Level 3 - E-move Level
   Inputs are symbolic names of "elemental" movements (E-moves), typically expressed as commands to achieve "key-frame" poses in coordinate system of choice.
5. Level 2 - Primitive Level
   Intermediate trajectory poses which define a path which has been checked for obstacles and is guaranteed free of collisions and kinematic singularities.
6. Level 1 - Servo/Coordinate Transfer Level
   Inputs are evenly spaced trajectory points for manipulators, grippers, transporters, and sensor platforms in a con-

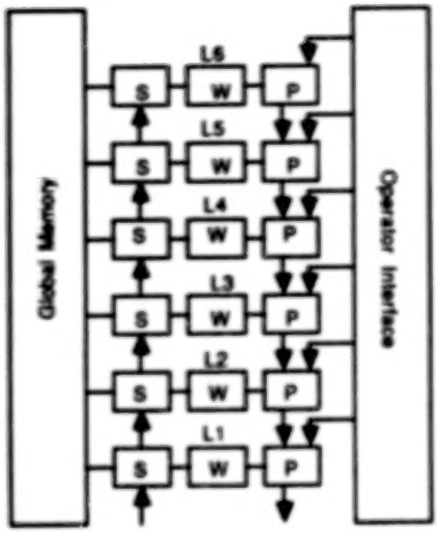


Figure 1: NASREM Telerobot Control Architecture

venient coordinate frame.

The output from Level 1 are commands to physical devices such as D/A converters. As one moves up the hierarchy the commands become less specific and more mission oriented.

At each level of the control hierarchy there are three different types of modules, the sensory processing module (S), the world modeling module (M), and the task decomposition system (H). The sensory processing system is used to sense physical processes in the FTS system. The world model uses this information to estimate the state of the system and to predict future states. Also, the world model is used to evaluate plans which are produced by the task decomposition system.

The operation of the task decomposition system at each level is characterized by an H-module. These modules convert higher level task descriptions into sequences of lower level subtask. Thus, the operation of the controller is basically defined in terms of the operation of the task decomposition system, or H-modules. Each H-module has three components: the Job Assignment, JA, which partitions the input task command into distinct jobs to be performed by physically distinct mechanisms; Planners, PL, which convert each job created by JA into a sequence of subtasks; and the Executors, EX, which execute the subtasks created by the planners.

2.2 Extensions to NASREM

To handle the problem of remoteness of the robots, we have divided NASREM into two parts which are called: Local NASREM, and Remote NASREM, Figure 2. The remote component is identical to standard NASREM except that the operator interface is replaced by a remote communications link. The local component contains the operator interface as well as an additional world modeling system and a local planning system. A significant difference is the addition of the world modeling system at the local site which is used for planning. This addition is important as the long communication delays imply more recent knowledge about the robot's world at the remote site than is available for planning at the local site.

Communication to the remote site can occur at any level down to level 3 of the hierarchy. Below that level the feedback loops are of such a high bandwidth to preclude closing the loop at the local site.

The final difference is the introduction of the Level 0 controller to indicate hard wired systems below Level 1 of the NASREM hierarchy. For example, a Level 0 controller is used to control the torque produced by the manipulator actuator.

The function of levels 0 through level 3 at the local site is to simulate the future operation of the remote site at levels 0 through 3 for the purpose of operator evaluation of plans produced by level 4 at the local site. In the current system, only level 3 of levels 0 through 3 at the local site are in place. Thus, the operator can view motions planned before the commands are sent to the remote site. However, because of the lack of the remaining levels at the local site, the operator can not preview forces which will be generated at the remote site.

The current configuration of the system is shown in figure 3. Only those components drawn in solid have been implemented. The following sections describe the design of these components of the system.

3 Local - Level 4

The language TROL (TeleRObot Language) was developed for operator input to Level 4. It was implemented using the pacc compiler-compiler which is available on all UNIX systems. The language provides a simple way of commanding motions to the robots.

The problems we have encountered are mainly the result of our limited view of the functionality of the robots at the remote site. In developing a robot programming language one is inclined to define a language in which only the motion of a robot can be specified. In retrospect, the language should specify the the way in which changes are to be made in the robot's environment. The language should only implicitly specify the motion of the robot. This would not only make it easier to integrate higher levels in the control hierarchy, but it would also simplify the operation of the servo controller at the remote site. As discussed in section 6 several control laws are available for selection. Examples include adaptive and compliant motion control laws. The appropriate selection should be based upon the particular task to be performed by the manipulator. In other words, the type of changes which are to be made in the robot's environment.

4 Local - Level 3

Level 3 of the hierarchy converts key-frame poses obtained from level 4 into sequences of positions which are free of kinematic singularities and collisions. We have found there are few solutions to this problem, and they are all computationally intensive [2].

It is much easier to determine if a given path is free of collisions than it is to numerically determine a collision free path. Therefore, our approach to this problem is to display the pose of the system at each key-frame pose. After viewing these po-

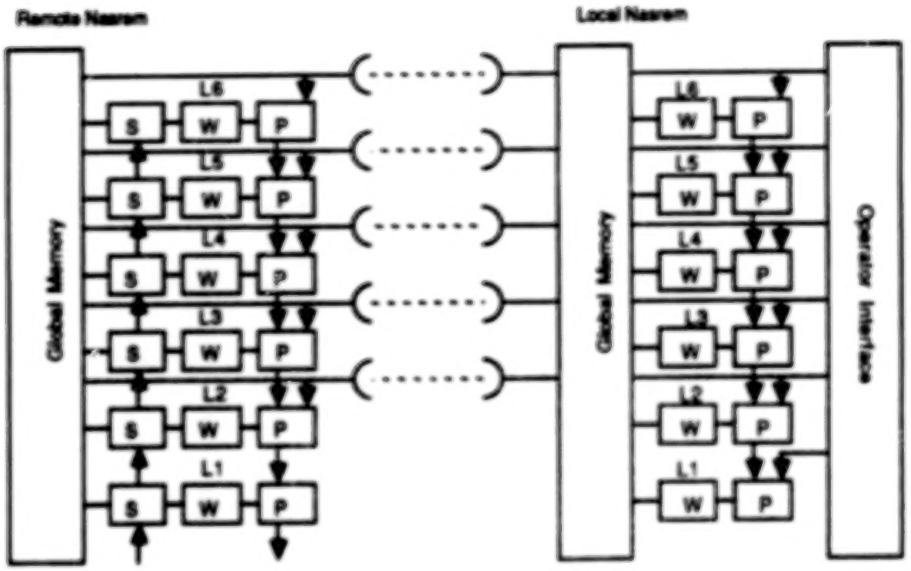


Figure 2: Extended NASREM

sitions, if a collision has occurred, the operator is compelled to create a new plan. He has to reenter a new program at level 4. Thus, the difference between the operation of our system and the NASREM system is the requirement of human interaction to provide collision free paths.

5 Remote - Level 2

Level 2 control is a simple a path interpolator [3]. Trajectories are composed of sequences of straight line segments connected together by smooth arcs. The positions obtained from Level 3 specify the end points of these line segments. For a trajectory having m segments, the $m-1$ intersection points of every two adjacent segments are denoted by the 6×1 vectors $\boldsymbol{p}_i$, $i = 1..m-1$. Accordingly, $\boldsymbol{p}_0$ and $\boldsymbol{p}_m$ denote the initial and final points, respectively.

During the execution of the $i-th$ segment, the manipulator is either in transition from the $i-1st$ segment, or on the $i-th$ segment, or in transition to the $i+1st$ segment.

The acceleration is choosen to be zero when the manipulator is in the middle of a segment and a constant non-zero value during transition to or from an adjacent segment. Since the acceleration is zero along the middle portion of a segment, the velocity, $\boldsymbol{v}_i(t)$, is constant. For example, in the middle of the $i-th$ segment

$$\boldsymbol{v}(t) = \boldsymbol{v}_i$$

where

$$\boldsymbol{v}_i = (\boldsymbol{p}_i - \boldsymbol{p}_{i-1})/T_i$$

T_i is the time duration required to go from points $\boldsymbol{p}_{i-1}$ to $\boldsymbol{p}_i$. During the transition from one segment onto another the acceleration is

$$\dot{\boldsymbol{v}}(t) = \dot{\boldsymbol{v}}_i$$

where

$$\dot{\boldsymbol{v}}_i = (\boldsymbol{v}_{i+1} - \boldsymbol{v}_i)/(2\tau_i)$$

$2\tau_i$, the time required to accelerate between segments i and $i+1$.

The equation describing the position of the manipulator is:

$$\frac{d\boldsymbol{p}}{dt} = \boldsymbol{v} \quad (1)$$

$$\frac{d\boldsymbol{v}}{dt} = \dot{\boldsymbol{v}} \quad (2)$$

where $\dot{\boldsymbol{v}}$ is either the null vector, corresponding to the time interval where the hand is in the middle of a segment, or $\dot{\boldsymbol{v}}_i$ corresponding to the time interval when the hand is in transition from the $i-1st$ segment.

The position and velocity of the hand are computed as a function of time by integrating Equations 1 and 2. Since the acceleration during any one sample period is always constant, this equation can be most easily solved by converting it into a sampled data equation. The resulting equations are:

$$\boldsymbol{p}(k+1) = \boldsymbol{p}(k) + \boldsymbol{v}(k)\Delta t + 1/2\dot{\boldsymbol{v}}(k)(\Delta t)^2 \quad (3)$$

$$\boldsymbol{v}(k+1) = \boldsymbol{v}(k) + \dot{\boldsymbol{v}}(k)\Delta t \quad (4)$$

where k represents the $k-th$ sample time, and Δt is the sample period.

Although Equations 3 and 4 are exact equations for computing $\boldsymbol{p}(k)$ and $\boldsymbol{v}(k)$, the procedure does suffer from round off errors and will tend to drift from the correct position and velocity. Thus, the numerical values of $\boldsymbol{p}(k)$ and $\boldsymbol{v}(k)$ are corrected for errors at the beginning of each straight line segment.

Note that the acceleration, velocity and position along the initial starting and the final stopping segment can also be specified by using this method. To do this, an additional segment of zero length is appended to the beginning and another to the end of the trajectory. This results in an initial and final velocity of zero and a start up time of $2\tau_1$ and a stopping time of $2\tau_{m-1}$.

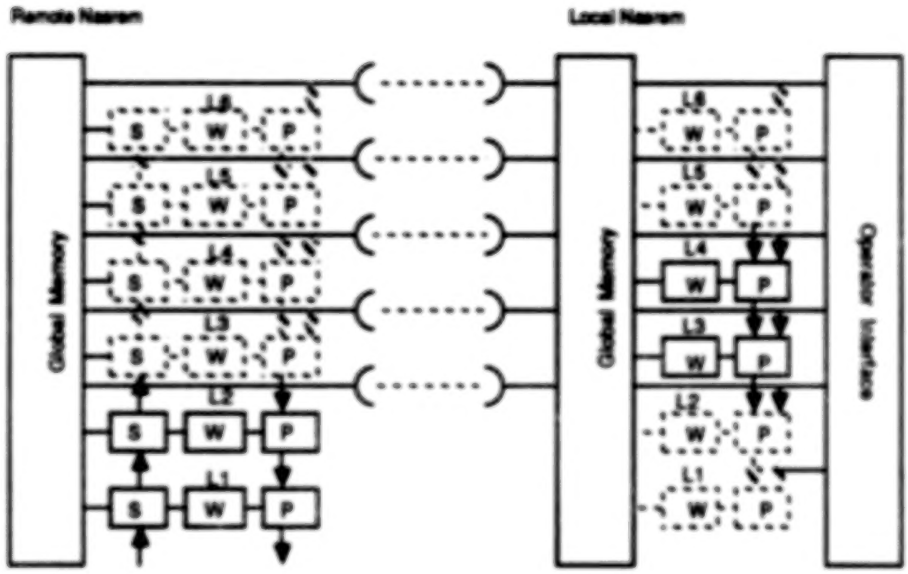


Figure 3: Current System Configuration

6 Remote - Level 1

The manipulator is modeled as a graph structured system consisting of $m+1$ links, $r \geq 0$ closed kinematic loops, and n primary joints whose motions are independent of the remaining joints. Note that if $r = 0$ then $n = m$, and the manipulator is a serial link manipulator. However, we have found that true serial link manipulators are nearly nonexistent. Even the PUMA 560 manipulators we use, which are usually considered serial link manipulators, are really graph structured systems, due to the internal gearing.

We use the spatial notation popularized by Featherstone, [4, 5, 6] for the modeling and control law formulation. This notation is convenient not only for analysis purposes, but also in simplifying the implementation of the controller. Since Ada was used in writing the software of the controller, we can overload the operators such as multiplication and addition to apply to the spatial quantities defined by Featherstone. Thus, the syntax of the software developed for the controller is very similar to the notation used in the analysis of the robot dynamics and control.

6.1 Kinematic Constraint Equations

Since, in general, the manipulator is graph structured, there exist constraints in the relative position of each of the joints of the manipulator. These constraints can be represented by a set of equations of the following form:

$$q = U(Q) \tag{5}$$

where Q is an $n \times 1$ vector containing the positions of the primary joints. The primary joints are a set of joints such that their position uniquely determines the position of all other joints in the manipulator. Note that, for each Q_i there is a q_j such that, $Q_i \equiv q_j$.

Since there are n primary joints, there must be at least n actuators in the manipulator. If the manipulator has more than n actuators it will be redundantly actuated [7]. That is, there would be more actuators than are required for motion. Although it is acceptable to choose the primary joints to be those where the motor armatures are located, it is not a requirement. Other joints are usually better. For example, the six primary joints of the PUMA 560 manipulator are those located between the end-effector and the base of the manipulator. Thus, all solutions of the inverse kinematics problem for the PUMA 560 have been presented for these joints of the manipulator. However, motor number four is kinematically coupled to joints four, five and six. Thus, changing any of these joint positions causes motion of the fourth motor.

Given $U(Q)$, $\dot{Q}$, and $\ddot{Q}$, the velocity of all the joints can be determined.

$$\dot{q} = E(Q)\dot{Q} \tag{6}$$

where

$$E(Q) = \frac{\partial U(Q)}{\partial Q} \tag{7}$$

The acceleration is given by:

$$\ddot{q} = E(Q)\ddot{Q} + \dot{E}(Q)\dot{Q} \tag{8}$$

For a given manipulator these equations are usually fairly simple. Often the matrix $E(Q)$ is constant. For example, it is a constant 6×6 matrix for the PUMA 560 manipulator. However, for some manipulators these equations can become complex. For these manipulators we have found that the c-algebra is very convenient for evaluating the time derivatives of the joint positions [8]. Using this algebra one only has to program the solution to Equation 5, change the order of the algebra and automatically obtain the solution to equations 6 and 8.

6.2 Feedback

Feedback signals used by the control come in two forms: spatial feedback vectors, $\bar{F}_j$, which are associated with the links of the manipulator and scalar feedback functions, ϕ_j, which are associated with the joints in the manipulator. These can be any functions of the desired state and the actual state of the manip-

ulator. The method used to calculate the required primary joint forces is as follows.

1. The values of the spatial feedback vectors, $\hat{F}_j$, are computed for each link, and feedback functions, ϕ_j, are computed for each joint of the manipulator.

2. The primary joint forces, τ_i, are computed using the $\hat{F}_j$ and ϕ_j as in the equation:

$$\tau_i = \sum_{j=1}^{n+r} (\epsilon_{ji}\phi_j + s'_{ji}\hat{F}_j)$$

where ϵ_{ji} is the $ji-th$ column of the matrix $E(Q)$, and s_{ji} is the spatial velocity of link j due to a unit velocity of the $i-th$ primary joint. An algorithm for the efficient evaluation of the τ_i has been presented, [6].

Thus, the controller simultaneously encorporates feedback at the joint level and at the Cartesian level. The choice of the $\hat{F}_j$ and ϕ_j dictate the character of the controller being implemented. For example, adaptive controllers for a single manipulator [6] and for dual arms [9] have been presented.

An important property of this controller is that regardless of the choice of these feedback functions, they all satisfy the following property.

$$\sum_{j=1}^{n+r} (\theta'_j(F_j + I_j g) + \dot{q}_j \eta_j) = \sum_{j=1}^{n+r} (\theta'_j \hat{F}_j + \dot{q}_j \phi_j) \qquad (9)$$

where F_j is the actual inertial force on link j, I_j is the spatial moment of inertia matrix, g is the spatial acceleration due to gravity, η_j is the joint j friction force, the $\dot{q}_j$ are any joint velocities which are consistent with the constraint equations and θ_j is the resulting link spatial velocity. This property is very useful for showing stability of the some feedback functions. Basically, it is simply a restatement of the principle of virtual work. If the $\dot{q}_j$ are the same as the actual joint velocities, $\dot{q}_j$, then the term on the left is the rate at which energy is absorbed by the manipulator and the term on the right is the rate at which energy is delivered to the arm by the actuators. In fact, by thinking of the $\hat{F}_j$ and ϕ_j as inputs to the system, equation 9 is simply another form of the equations of motion. That is, the equations of motion are:

$$\sum_{j=1}^{n+r} (s'_{ji}(F_j + I_j g) + \epsilon_{ji}\eta_j) = \sum_{j=1}^{n+r} (s'_{ij}\hat{F}_j + \epsilon_{ij}\phi_j) \quad i = 1 \ldots n$$

Thus, Level 1 of the controller calculates the spatial feedback functions $\hat{F}_j$ and the scalar feedback functions ϕ_j, converts them to equivalent primary joint torques, τ, and outputs τ to the Level 0 controller. As described in the next section, the Level 0 controller converts these to the equivalent motor actuator torque, and then controls the current in the motor to attain that desired torque.

The software of the controller is organized so that adding new feedback functions only requires the user to add two new procedures called, **forward** and **backward**. At each sample period, these procedures are called once for each link of the manipulator. The procedure **forward** is used to update any local state information such as local digital filters that are used by the feedback function. The procedure **backward** evaluates and returns the values of the spatial feedback functions $\hat{F}_i$ and the scalar feedback functions ϕ_i. Again, the particular feedback function used is dependant upon the task being performed by the manipulator. For example, adaptive control requires different feedback than compliant motion control.

7 Remote - Level 0

Level 0 is the hard wired control of physical devices to make them look like the model of the devices used in the formulation of the controller at level 1. For the case of the manipulator controller, all level 1 controllers assume they are controlling a manipulator with motors at the primary joints. However, because of the use of gearing in most manipulators, The motor armatures are not located at the joints chosen as primary joints. The primary joint positions and the motor armature positions are related by the following equation.

$$Q = R(q_m)$$

where q_m are the motor armature positions. Given τ from the level 1 controller, the level 0 controller converts this to an equivalent motor torque by the equation:

$$\tau_m = B(q_m)^T \tau$$

where $()^T$ indicates the transpose of the matrix and

$$B(q_m) = \frac{\partial R(q_m)}{\partial q_m}$$

The level 0 controller then drives the current through the DC motor used for actuation to produce the desired motor torque. There are two types of motor actuation systems that could be used. The PUMA 560 manipulator is an example of the first type. In this system the only sensory signals are the current flowing through the armature and the position of the motor. It is an indirect control of the torque produced by the motor. The torque is controlled by directly controlling the currents through the motor armatures. The torque is the assumed proportional to this current. The Robotics Research Arm [10] is an example of the second type. This arm has a sensors to measure the output torque produced motor in addition to the motor position. The control of torque in the Robotics Research Arm is a direct control since the output torque is directly measured and fed back to the input of the motor.

8 Hardware

The connections to the Level 0 control are through the same umbilical utilized by the standard PUMA 560 to connect to the VAL II system controller. Thus, the Level 0 hardware, in effect, replaces the controller hardware which comes standard with the PUMA 560. Between the PUMA 560 and the Level 0 controller hardware is the robot interface box. This box contains all the electronics specific to the manipulator being controlled. It contains the arm power supply, the joint amplifiers, the encoder signal conditioning circuitry, and the brake release circuitry. Digital signals between a VME system and the robot interface box use differential transmitters and receivers for noise immunity, and to allow a large distance between the two systems. The joint amplifiers are PWM-type power amplifiers. The signals from this box are then directly connected to the digital I/O, and D/A and

A/D converts resident in the VME chasis. A Motorola 68020 based single board computer is used to implement the Level 0 controller algorithms. All software written for this computer is in the *C* programming language. *C* was chosen because of the low cost of the compiler (Alycon Corp.), the speed of the *C* language and the simplicity and relatively small size of the software developed at this level.

Level 1 and 2 of the controller were implemented on a Compaq 386/20 computer using Ada as the programming language. The interface between Level 1 and Level 0 controls is through a 64k shared memory block. Thus, the Compaq simply writes out torque commands by writing to a specific location of the shared memory block and the Motorola 68020 presents the sensory information such as primary joint positions and velocities by writing to the same shared memory block. Thus, a very fast communication mechanism is used between these two levels.

The world modeling system and levels 3 and 4 of the local system reside on a Silicon Graphics 4D-GT graphics work station. This graphics work station is fast enough to simultaneously present a graphic display of the current operation of the remote robots and also a second display of the plans that are being developed for future operation of the robots. The only real video that is used is though a video telephone, which transmits pictures at about a 5 second frame rate. The real video is only used by the operator for verification of the operation of the remote robots.

9 Remote Communication

The current system allows commands to be sent to the remote robots via two transmission mediums : the Internet Network and the common telephone line. This section describes this implementation.

9.1 The Internet Transmission Control Protocol

For hosts on the Internet Network, the Internet Transmission Control Protocol (TCP) provides a convenient medium of passing commands between the manipulator and the controlling agent. At its highest level of abstraction, the TCP protocol provides a potentially reliable, sequenced, full-duplex connection-based byte-stream. This communication protocol is referred to as the socket stream. A socket stream must be connected before any data can be sent or received on it. Hence, the controlling agent who wishes to control the manipulator, must first make a request for connection. If the manipulator grants the request, then a full-duplex stream of commands may be passed between the two agents. These commands are encoded into fixed sized command packets and then sent over the socket stream. The agent on the opposite end of the socket stream, will then decode the command packet and take the appropriate action. When the session is over, both agents will close their end of the socket stream, and the manipulator will reset and listen for further connection requests.

9.2 Serial Socket

For those controlling agents not on the Internet Network, a method of communication over the standard phone lines was developed. At the heart of this serial communication is a protocol dubbed the *serial socket*. The serial socket is a protocol which implements socket-like attributes over a standard RS-232 serial line. Specifically, the serial socket provides a potentially reliable, sequenced, full-duplex byte stream over a serial line, or with the aid of a pair of modems, over a standard telephone line. Similar to the socket stream, the serial socket must be connected before any data can be sent or received on it.

The controlling agent requires minimal hardware and software :

- A computer running Unix BSD 4.2 or Unix System V.
- A 2400-baud Hayes-compatible modem supporting the full AT-command set.
- The serial socket and associated software.
- A reliable phone line.

Connection with the manipulator is achieved in a similar manner as the socket stream, except that the controlling agent must first utilize the serial socket software to dial the phone number of the telephone at the remote site. If the connection request was granted, then a full-duplex stream of commands may be passed between the two agents over the serial socket.

As mentioned above, the serial socket shares the high level attributes of the socket stream. To achieve this level of abstraction, a three layer communication protocol was developed. The lowest layer of the protocol directly interacts with the serial port. Routines in this layer set the baud rate of the port, read bytes from the port, write bytes to the port, etc. Similar to the Kermit protocol, the serial socket library makes only minimal assumptions about the serial port over which the transfer occurs; namely that the port is capable of sending and receiving all printable ASCII characters. It also requires that the system be able to send and receive a SOH control character. Most Unix systems provide this facility by incorporating the serial port as a special file.

The middle layer in the serial socket communication protocol involves the transmission and reception of fixed size packets over the serial port. By default, the size of these packets is 128 bytes. Included in the packet are fields for the packet type, sequence number, encoded data, and check value. To meet the requirement mentioned above, only printable ASCII characters are allowed to reside in the packets. To this end, the binary data is encoded to a purely printable form when a packet is written to the serial port, and decoded back to binary when it is read. This mapping is the same as that used in the Kermit file transfer program. To detect errors during the transmission of a packet, a check value field is included in each packet. By default, a 16-bit Cyclic Redundancy Check (CRC-16) is used. The CRC is good at detecting all kinds of errors (single-bit, double-bit, odd-numbered, etc), but especially those that occur in bursts over a relatively long time.

The topmost layer in the serial socket communication protocol implements the automatic repeat request (ARQ) packet protocol. An error detected in a received packet or an unacknowledged packet automatically results in the retransmission of that packet. A high level description of this protocol follows. During transmission, the binary data is packetized by surrounding it with service fields. The entire packet is then transmitted with no flow control, after which the sender waits for the receiver to acknowledge its receipt. The receiver inputs the packet and, after

verifying that the packet is in the correct sequence, computes a local checkvalue on the data portion of the packet. If this checkvalue matches the one in the packet, the receiver acknowledges by sending an acknowledge (ACK) packet. If the packet was in error, then a negative acknowledge (NAK) packet is transmitted. Upon receipt of an ACK, the sender transmits the next packet; if a NAK was received, then the same packet is transmitted again. Transmission proceeds in this manner until the serial socket is closed.

10 Conclusion

The design of a testbed for the control of a remote robotic system has been presented. The NASREM control architecture was used as a basis for the system development. Because of the remoteness of the robots some extensions to the NASREM architecture were proposed.

In the current operation of the system we have observed that the limitations of our system basically stem from the deficiencies of our communications language and the *side effects* of commands.

The communications problem stems from our basing the communications scheme on an enumerate of all possible commands to the remote system. We are now in the process of designing a system which uses the Ada programming language as the communications language. Thus, we are, in effect, using an Ada interpreter for the implementation of level 4 at the remote site. This will give us the capability of doing such things as defining a variable for the location of an object and then referring to the location of the object through the name of that variable. Thus, the local site need not know the exact location of the object. It can refer to this exact location which is stored at the remote site through the variable it defined for that purpose. Also, by using Ada as a basis for communications we will derive all the benefits of a well defined language. Any person who knows Ada could program the robot.

The other main problem with our control is the problem with side effects. That is , many commands which are issued to the remote site may or may not have side effects which are different that the primary effect intended for that command. An example is the command *Close Gripper*. The primary effect of this command is to actuate the fingers which may or may not cause something to be grasped. If something is grasped, then there will be a side effect in future motions of the end-effector. Specifically, the position of the grasped object will change with a change in position of the end-effector. In addition, if the object is attached to another object, as for example, a door handle connected to a door, then as the manipulator end-effector is moved, the location of the door is moved. Whats more, this side effect places constraints on the motion of the end-effector. The controller designer must include these constraint effects in his controller design.

Our current effort is directed towards providing a communications language which better describes the desired changes in the robots environment and to modify the robot controller to carry out those desired changes.

Acknowledgement

This work was supported by a grant from the NASA sponsored Center for Autonomous and Man-Controlled Robotics and Sensing Systems at ERIM, Ann Arbor, MI.

References

[1] J. S. Albus, H. G. McCain, and R. Lumia, "Nasa/nbs standard reference model for telerobot control system architecture (nasrem)," Technical Report NASA/SS-GSFC-0027, National Bureau of Standards, March 13 1987.

[2] T. Lozano-Perez, "A simple motion-planning algirithm for general robot manipulators," *IEEE J. Robotics and Automation*, vol. RA-3, pp. 224–238.

[3] R. P. Paul, *Robot Manipulators: Mathematics, Programming and Control*, MIT Press, Cambridge, Mass, 1981.

[4] R. Featherstone, "The calculation of robot dynamics using articulated-body inertias," *The Int. J. of Robotics Res.*, vol. 2, no. 1, pp. 13–30, Spring 1983.

[5] R. Lathrop, "Constrained (closed-loop) robot simulation by local constraint propagation," In *I.E.E.E. Int'l Conference on Robotics and Automation*, San Francisco, CA, 1986.

[6] M. W. Walker, "An efficient algorithm for the adaptive control of a manipulator," In *I.E.E.E. Int'l Conf. on Robotics and Automation*, Philadelphia, Pennsylvania, April 1988.

[7] J. Gardner, V. Kumar, and J. Ho, "Kinematics and control of redundantly actuated closed chains," In *I.E.E.E. Int'l Conf. on Robotics and Automation*, Scottsdale, Arizona, May 1989.

[8] M. W. Walker, "Manipulator kinematics and the epsilon algebra," In *I.E.E.E. Int'l Conf. on Robotics and Automation*, Raleigh, North Carolina, 1987.

[9] M. W. Walker, D. Kim, and J. Dionise, "Adaptive coordinated motion control of two manipulator arms," In *I.E.E.E. Int'l Conf. on Robotics and Automation*, Scottsdale, Arizona, May 1989.

[10] "Specification for type 1 & type 2 motion controller servo-level interface," Technical Report OPS SLI-02888-1A, Robotics Research Corporation.

TECHNOLOGY FOR AN INTELLIGENT, FREE-FLYING ROBOT FOR CREW AND EQUIPMENT RETRIEVAL IN SPACE

J. D. Erickson, G. J. Reuter, K. J. Healey
NASA Johnson Space Center Houston, Texas 77058
and
D.E. Phinney
Lockheed Engineering and Sciences Company Houston, Texas 77058-3711

ABSTRACT

Crew rescue and equipment retrieval is a Space Station Freedom requirement. During Freedom's lifetime, there is a high probability that a number of objects will accidentally become separated. Members of the crew, replacement units, and key tools are examples. Retrieval of these objects within a short time is essential.

Systems engineering studies were conducted to identify system requirements and candidate approaches. One such approach, based on a voice-supervised, intelligent, free-flying robot was selected for further analysis. A ground-based technology demonstration, now in its second phase, was designed to provide an integrated robotic hardware and software testbed supporting design of a space-borne system.

The ground system, known as the EVA Retriever, is examining the problem of autonomously planning and executing a target rendezvous, grapple, and return to base while avoiding stationary and moving obstacles. The current prototype is an anthropomorphic manipulator unit with dexterous arms and hands attached to a robot body and latched in a manned maneuvering unit. A precision air-bearing floor is used to simulate space. Sensor data include two vision systems and force/proximity/tactile sensors on the hands and arms.

Planning for a shuttle flight experiment is underway. A set of scenarios and strawman requirements were defined to support conceptual development. Initial design activities are expected to begin in late 1989 with the flight occurring in 1994. The flight hardware and software will be based on lessons learned from both the ground prototype and computer simulations.

Presented at Space Operations Automation and Robotics Conference and Workshop, JSC, Houston, TX, 77058, July 25-27, 1989.

INTRODUCTION

A requirement exists to provide a retrieval capability for objects (astronauts, equipment, and tools) which have separated from Space Station Freedom. An analysis of the amount of crew Extra Vehicular Activity (EVA) likely during the lifetime the Space Station indicates, with high probability, that a number of objects will accidentally become untethered. Crew safety is top priority. In addition equipment may be too valuable to lose because it is required in operations and replacement is not available on the station. There is also collision potential on later orbits which, though small, has occurred previously.

The Space Station itself will lack the capability to chase separated crew or equipment and other vehicles such as the Space Shuttle orbiter or the Orbital Maneuvering Vehicle will not usually be available. Potential solutions based on manned, teleoperated, and autonomous capabilities have all been proposed.

Retrieval by a crew member using a Manned Maneuvering Unit (MMU) was examined in some detail. Analysis revealed that a short response time is critical. Many hours of real-time simulation of retrievals indicated that manned retrievals were unlikely to provide the required response time. In any case a major and unacceptable risk to the astronaut was involved.

The evolving requirements call for an unassisted deployment from a mounting on the external part of the airlock with propulsion capabilities provided by a more powerfull version of the existing MMU. Performance guidelines include target retrieval within 120 minutes of subsystem deployment. Reliablity considerations mandate the use of fault tolerant and fail-safe designs with embedded fault detection and isolation capabilities. Safety, reliability, robustness, and maintainability in space are key attributes.

Space Station Freedom advanced automation and robotics has been the subject of numerous symposia and papers [1, 2]. Appropriate roles for humans and machines in an evolving mix have been highlighted as a specific goal, with supervised intelligent system designs as ways to meet the needs of appropriate flexible-capability automation and robotics, thereby giving people-amplifier-type productivity gains.

The retrieval problem provides an opportunity to evaluate such systems in the form of a supervised, intelligent, free-flying space robot. The concept of supervised, intelligent, autonomous robotics provides for autonomous behavior of an intelligent type where human control is normally at a high level of goal-setting and involved in mixed initiative communication as a means of implementing decentralized, delegated management. By contrast, telerobotics provides a partially automated remote extension of human task performance with occasional control delegation for specific parts of tasks given to the telerobot for efficiency reasons.

Several previous efforts have laid a foundation for autonomous robot development including Shakey [3], JASON [4], the RPI Rover [5], the JPL Rover [6], and the Stanford Cart [7], among others. These first-generation autonomous robots were used to explore basic issues in vision, planning, and control. However, they were all seriously hampered by primitive sensing and computing hardware.

More recent efforts have overcome many of these limitations, and very sophisticated second generation autonomous robot testbeds have evolved. Some of these efforts include the developments of HILARE [8], the FMC Autonomous Vehicle [9], the Autonomous Land Vehicle (ALV) [10], the various CMU mobile robots [7], and the Ground Surveillance Robot (GSR) [11]. A more general and complete discussion of autonomous vehicle history and technical issues has been given by Harmon [12]. While operational versions don't exist, much advantage can be obtained from these efforts.

By comparison, the space retrieval task seems simpler in some respects. While automatic control, such as is available in automatic guided vehicles (AGV), remotely piloted vehicles (RPV), and missiles, is not adequate here due to the dynamic environment, the more general solutions to vision and planning in completely unknown environments are not required. There are few objects in space; these are cooperative, and largely knowable. In low earth orbit, space is characterized by high thermal gradients, radiation levels, high vacuum, microgravity and reaction-force aspects, and constrained and delayed access to information, resources, and equipment. Supervision by voice is a natural, flexible means of providing the primary human-machine interface (supplemented with helmet displays) required. This requires limited natural language understanding integrated with the environment and task as well as functions like planning and reasoning. Complete intelligent autonomy of the R2D2/C3P0-type is not required nor achievable.

However, significant technology advances will be necessary before even this simple, crucial application can be practically addressed. These advances will only be gained by implementing autonomous robot simulations and testbeds so as to gain experience with the developing technology.

The potential evolution of such a robot to an EVA crew helper is obvious. Routine inspections, fetching tools, holding objects, could all improve EVA safety and productivity.

The EVA Retriever ground-based technology demonstration study [13, 14] was established to design, develop, and demonstrate in three phases an integrated robotic hardware/software system which supports design studies of a space borne crew rescue and equipment retrieval capability. Goals for each phase were established [15] in support of the overall goal of building and evaluating the capability to retrieve objects (astronauts, equipment, and tools) which have accidentally separated from their spacecraft. The Phase I goals were to design, build, and test a retriever system testbed by demonstrating supervised retrieval of a fixed target. Phase II goals are to initiate simulations and to enhance the testbed subsystems with significant intelligent capability by demonstrating target retrieval while avoiding fixed, arbitrarily oriented obstacles. Phase III goals are to more fully achieve supervised, intelligent, autonomous behavior by demonstrating retrieval of a moving target while avoiding moving obstacles.

Space Station scenarios [16] were examined in some detail to aid in the definition of a set of design reference missions. A number of systems engineering studies were conducted in support of the software design. Level A requirements for a projected Space Station version were developed in a conceptual design study [17]. Level B software requirements were derived in greater detail for this possible future Space Station application [18].

This paper gives an overview of the experimental hardware and software and a brief summary of the Phase II experiment. These are related to the current planning for a shuttle flight experiment.

PHASE II PROTOTYPE

The technology demonstrations are being conducted on the JSC Precision Air Bearing Floor (PABF). The retriever/MMU unit is mounted on a test stand with

compressed air supplied thru an umbilical. The MMU has twenty-four thrusters, four on each rectangular side. The MMU accepts simple translation and rotation on/off commands to fire thrusters providing fixed acceleration in any of the three translational or three rotational directions.

The current prototype (Figure 1) is an anthropomorphic manipulator unit with dexterous arms and hands attached to a robot body and latched in an MMU. Sensor data include accelerometers, gyroscopes, two independent vision systems, and force/proximity sensors on the hands and arms. The primary vision system consists of a laser scanner imager and video camera mounted on a controllable turntable. The secondary vision system is a multicamera video tracking system, with a chest camera array and a camera in one of the hands.

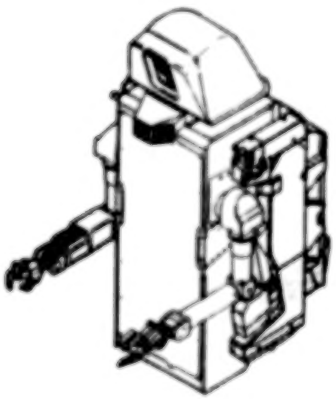


Figure 1. Retriever test article.

The prototype has dual 6-degree-of-freedom arms. The arms have roll and pitch at the shoulder, elbow, and wrist. One of the arms has a dexterous grasping hand and the other has a three fingered gripper. The dexterous hand incorporates proximity sensors in order to support adaptive grasping of an object by monitoring force and moment buildup.

The processor configuration contains seventeen transputers (five of which are dedicated to vision processing), several 68020 processors, a 80386 processor, and a special purpose video tracker subsystem.

The EVA Retriever software is required to autonomously plan and execute a target rendezvous, grapple, and return to base while avoiding stationary and moving obstacles. The system is required to monitor plan execution, estimate probability of mission success, and dynamically replan whenever needed to achieve system goals.

The software architecture (Figure 2) incorporates a hierarchical decomposition of the control system that is horizontally partitioned into five major functional subsystems: perception, world model, reasoning, sensing, and acting. The design utilizes hierarchical flow of command and status messages but allows horizontal flow of data between components at the same level. Computation is performed at the lowest possible level and, in general, knowledge-based systems are utilizated only when algorithmic solutions are lacking in power or flexibility. This approach handles multiple levels of abstraction well and permits the incorporation of special data paths between time critical components.

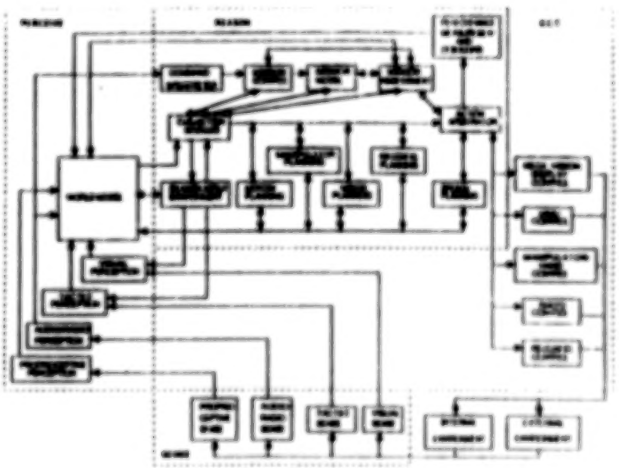


Figure 2. EVA Retriever software components.

The overall design provides for an evolutionary system improving in capability over time and as it earns crew trust through reliable operation. Additional details on the hardware and software design may be found in Erickson et. al. [14].

PRELIMINARY PHASE II RESULTS

At the time of this writing, hardware and software integration is nearing completion in preparation for PABF evaluations and demonstrations in the summer of 1989. Nevertheless, some preliminary results are available. Several static tests have been successfully completed which indicate successful integration and operation of subsystems.

The computer simulation testing carried out in Phase II for purposes of unit and integrated software dynamic testing of prototype designs has already paid major dividends. First, we have gained solid confidence in the Retriever's behavior and the software which drives this behavior. Second, we have been able to investigate

detailed requirements issues and design issues into which we otherwise would have little means to gain insight. Third, as a result, we have been able to make numerous minor requirements, design, and implementation changes and test them before hardware integration or PABF evaluations. So, as expected, the simulation testing has been very useful and efficient even though it is not the whole answer. Building the complete artifact and testing it in a set of physical tests including grappling cannot be simulated and is required to gain confidence in the design and understand its limitations.

In attempting to design realtime visual perception for grasping some preliminary results were obtained which relate to sensors and engineering of computer hardware and software for robotic applications. Images of moving objects taken with the laser scanner at 0.8 seconds per frame show a warped and distorted image of the objects for even reasonably slow moving objects. This result indicates that frame rates need to be increased substantially to deal with moving objects.

The range image processing from laser scanners done by Vemuri and Aggarwal (19) using curvature representations in Constructive Solid Geometry (CSG) approaches for stationary objects establishes, in principle, an algorithm for constructing the surface of an arbitrary, 3-D unknown man-made object and, in particular, demonstrates the need for spline fitting, which is computationally intensive. Even though this algorithm is parallel in each "patch" of the image, extrapolation from the Vax sequential implementation to a parallel transputer one still leaves a computational period of many minutes which is not near enough to realtime to be a practical solution for robotic grasping. A simpler representation with less computational requirements seems indicated and is being sought without loss of generality. There are also sensor requirements implied by this algorithm which means many more pixels from the object are needed than our current scanner provides (that is, the IFOV must subtend a much smaller angle which coupled with higher frame rates may require greater laser power).

The performance measurement and debugging function is difficult conceptually because defining good measures of performance is not easy, but some progress has already been made. The motivation is simple -- with measurements of performance occurring constantly, the Retriever, its supervisor, and its designers can know how it is performing. As we are focused at the moment on requirements and design, measurements of performance provide the data needed to quantify limitations and thus where design improvements will be most useful. One measure of performance is the time required for rendezvous for a given arrangement of obstacles and target. Another is the thruster durations during this rendezvous needed for translations and rotations. A third is the distance travelled. In space, minimum time and minimum fuel trajectories are of interest. On the PABF, "good" trajectories weigh distance, time, and fuel.

Preliminary measurements of time for rendezvous and total thruster durations for one distance and object/target scenario from testing against simulation gave about 130 seconds for rendezvous and 13 seconds total thruster duration for the case with translations and rotations (including the head) occurring in parallel. Measurements on the air bearing floor will be compared to these simulation results.

Debugging is another important operational function. Although we have used the best third-party debugging software we could find for a multi-processor transputer configuration, we do not yet have an adequate capability, meaning it takes too long to find bugs because the tools are not supportive enough.

Another result from simulation testing dealt with minor modifications to the design of the world model in reasoning about whether an object was an object seen before or a new object. This deals with robustness to inertial measurement drift in space when no spacecraft radar is available for tracking Retriever and multiple objects. The event can be described as an unsensed change of location of Retriever. On the PABF this can occur due to floor disturbances causing sliding not sensed by accelerometers or gyroscopes. We use vision derived object location to provide feedback to Retriever about Retriever location. However, when an object appears to be in a relative location where no object was seen the last look and far enough from where one was seen, we call it a new object. Our design is now more robust in this one respect due to simulation testing and subsequent design modifications.

Another testing/modification case of this same kind dealt with parallel actions, that's doing two or more things simultaneously rather than sequentially. In the situation where a new object is seen while on a planned rendezvous trajectory, motion continues until a new motion plan is computed, or if too close to impact, a hold is executed until a new plan is available. We expect reactive planning to further improve our responsiveness.

DYNAMIC ENVIRONMENT

Phase III of the Ground Demonstration Program was intended to deal with moving objects in a dynamic environment (Phase II dealt with stationary objects). Consequently, requirements, design, implementation, and evaluation of techniques for moving objects have been planned for 1990.

An overview of our technical approach to moving objects is simply that Retriever needs to formulate and execute plans using visual perception for object search, acquisition, recognition, tracking, and grappling. Retriever will formulate and execute plans for mobility with moving obstacle avoidance to rendezvous with the moving target. Retriever will use dexterous manipulation and a grappling mechanism for grappling, tethering, and transfer. The plans need to be adaptable to the specific situation and to compensate for unknowns with reactive plans which can tolerate failed actions or react to unexpected events.

Safety Policy in Software

A dynamic environment raises the issue of providing safety. Retriever must be safe to use as it carries out its retrieval tasks, which will vary, in the face of unintended contacts, failed actions (grapples), and mechanical failures. We mean by being safe that Retriever must not harm an astronaut, any part of another spacecraft, or itself. We will employ technology (to provide safety) which supports guarantees on robot behavior.

Our approach is to provide safety via two approaches. The first is software safety technology which is concerned with ensuring that software will execute in a systems context without resulting in states of unacceptable risk and will take actions to remove the Retriever from conditions of unacceptable risk if they should occur due to detectable hardware or software faults or command errors -- a policy that states that Retriever software will neither create nor ignore states of unacceptable risk. Risk is defined as danger times hazard severity where danger is the probability of a hazardous state leading to an accident and severity is the worst possible damage that could result. A hazards analysis for software is planned for 1990.

The second approach is to provide safety via reactivity where we will encode plans as networks of responses to possible situations and where guarantees on behavior are sought via a finite state machine analogy.

SHUTTLE FLIGHT EXPERIMENT PLANNING

Planning for a Shuttle flight experiment in 1994 is underway. The primary technical objective of the flight experiment is to develop and demonstrate from the STS base a flight prototype of a crew supervised, intelligently autonomous space robot for the retrieval of free-floating objects. A tentative schedule of activities required is shown in Figure 3.

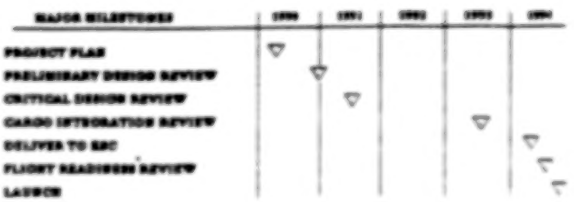


Figure 3. Preliminary Flight Experiment Schedule.

A preliminary set of scenarios were defined to support conceptual development. The primary demonstrations are the autonomous robotic retrieval of free floating targets in space (tool, astronaut model). The set of scenarios is organized in order from simple to more difficult to maximize success and minimize risks. There are essentially three groups of scenarios: those in the payload bay prior to commitment of EVA; those acquisition, tracking, and grappling tests in the payload bay with an EVA astronaut as supervisor; and free-floating target retrieval demonstrations well out of the Shuttle bay organized in order of increasing difficulty.

These latter demonstrations include a fixed target retrieval held by the Remote Manipulator System on an extension pole, a free- floating but stable tool retrieval, a rotating tool to be retrieved using a grappling mechanism (such as a net) , a tool retrieval with obstacle avoidance (no obscuration of target), a tool retrieval with temporary obscuration of the target by an obstacle, and an astronaut model retrieval.

Initial requirements and design activities are planned to begin late in 1989 for the actual flight in 1994. The necessary evaluations of approaches to deal with moving objects in a dynamic environment must be factored into the design. In fact, flight hardware and software will be based on lessons learned from both the ground prototype and a complementary series of computer simulations of moving objects. Flight system reliability will be better understood, crew training will be easier, and Shuttle integration will be less costly due to the use of the previously flown MMU.

The Crew and Equipment Retrieval Systems (CERS) is an identified part of the Space Station. A CERS study is currently underway by the Space Station Work Package 2 contractor, McDonnell Douglas. The study is currently in Part I which is a requirements definition

phase. Part I products will include a description of concepts evaluated and their programmatic impacts, and recommendation for retrieval capability implementation for both Station PMC and for Station Phase II. The Part I study will also define a study plan for the Part II CERS study. The Part II study would be utilized to produce at least two point design concepts. One concept would possibly be a simple astronaut self-help device for use by a conscious crew member in the early Station operation. The second design concept produced would be a free-flying retrieval system capable of satisfying all appropriate crew/equipment retrieval requirements.

The CERS study is directed to consider a supervised intelligent autonomous system such as the EVA Retriever and its related test programs. The EVA Retriever flight experiment would thus serve as a test bed for a variety of functions and equipment directly applicable to the Space Station CERS. This includes tracking, control, target recognition, grapple methods, crew control, and Station interface definition.

CONCLUSION

Evaluation of improved technology for the practical realization of a potential solution to the need for retrieval of crew and equipment in space near their spacecraft is underway. Preliminary results from the second phase of the ground testbed activity have been obtained from both computer simulations and the ground prototype. Important evaluations of technology to deal with a dynamic environment of moving objects including safety software are planned for 1990. Hardware and software lessons learned will be factored into planning for a Shuttle flight experiment. Assessment of practicality will rest on experimental evidence when these are completed.

ACKNOWLEDGMENTS

The EVA Retriever is a joint project of the Crew and Thermal Systems Division (CTSD), the Tracking and Communication Division (TCD), the Systems Development and Simulation Division (SDSD), the Avionics Systems Division (ASD), and the Structures and Mechanics Division (SMD), all of the Engineering Directorate at JSC, under funding from the JSC Director's Discretionary Fund. The contributions of the project teams encompassing civil servants and contractors to what is reported here are gratefully acknowledged.

REFERENCES

1. Cohen, A. and Erickson, J. D. "Future Uses of Machine Intelligence and Robotics for the Space Station and Implications for the U.S. Economy," IEEE Journal of Robotics and Automation, Vol. RA-1, No. 3, September 1985.

2. Erickson, J. D. "Intelligent Systems and Robotics for an Evolutionary Space Station," Journal of the British Inters," P Int. Joint Conf. Art. Intell., Washington, D.C., May 1969, pp. 509-520.

3. Nilsson, N., "A Mobile Automaton: An Application of Artificial Intelligence Techniques," Proc. Int. Joint Conf. Art. Intell., Washington, D.C., May 1969, pp. 509-520.

4. Smith, M., et al, "The System Design of JASON, A Computer Controlled Mobile Robot," Proc. IEEE Conf. Cybernetics and Society, San Francisco, CA, Sept. 1975, pp. 72-75.

5. Yerazunis, S., Frederick, D., and Krajewski, J. "Guidance and Control of an Autonomous Rover for Planetary Exploration," Proc. IEEE Milwaukee Symp. Auto. Computation and Control, Milwaukee, WI, April 1976, pp. 7- 16.

6. Miller, J. "A Discrete Adaptive Guidance System for a Roving Vehicle," Proc. IEEE Conf. Decision and Control, New Orleans, LA, Dec. 1977, pp. 566-575.

7. Moravec, H. "The Stanford Cart and the CMU Rover," Proc. IEEE, Vol. 71, July 1983, pp. 872-884.

8. Giralt, G., Chatila, R., and Vaisset, M. "An Integrated Navigation and Motion Control System for Autonomous Multisensory Mobile Robots," Proc. 1st Int. Conf. Robotics Res., Bretton Woods, NH, Aug. 1983.

9. Nitao, J. and Parodi, A. "A Real-time Reflexive Pilot for an Autonomous Land Vehicle," IEEE Control Sys. Mag., Vol 6, Feb. 1986, pp.14-23.

10. Lowrie, J., et. al."Autonomous Land Vehicle," Annual Report, MCR-85-627, ETL-0413, Martin Marietta Aerospace, Denver, CO, Dec. 1985.

11. Harmon, S. "The Ground Surveillance Robot: An Autonomous Vehicle Designed to Transit Unknown Terrain," IEEE J. of Robotics and Auto., Vol. RA-3, No. 3, June 1987, pp. 266-279.

12. Harmon, S. "Autonomous Vehicles," in Encycl. of Art. Intell., New York, Wiley, 1986.

13. Reuter, G. J., Hess, C. W., Rhoades, D. E., McFadin, L. W., Healey, K. J. and Erickson, J. D. "An Intelligent, Free-flying Robot," SPIE Symposium on Advances in Intelligent Robotic Systems, Space Station Automation IV, Cambridge, MA, Nov. 6-11, 1988.

14. Erickson, J. D., Phinney, D. E., Norsworthy, R. S., Zacksenhouse, M., Hartness, K. T., Pham, T. T., Merkel, L. F., and Tu, E. Y. "A Prototype Autonomous Agent for Crew and Equipment Retrieval in Space," Second International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, University of Tennessee Space Institute, June 6-9, 1989.

15. EVA Retriever Program Plan, Johnson Space Center, Houston, TX, JSC-22144, May 1987.

16. Shapiro, D. "An EVAR Scenario," Advanced Decision System, Mountain View, CA, June 1988.

17. Freeman, J."Autonomous Robot Control System Design Study," Ford Aerospace Corp., Houston, TX, JSC-12421, May 1988.

18. Taylor, D. et al, "Level B Post-PMC Space Station EVAR Functional Requirements, (Draft)," Project Report, Houston, TX, March 1988.

19. Vemuri, B.C. and Aggarwal, J.K., "3-D Reconstruction of Objects from Range Data," Intl. Conf. Pattern Recognition, Montreal, Canada, August 1984, pp.752-754.

THE JPL TELEROBOT OPERATOR CONTROL STATION - OPERATIONAL EXPERIENCES

Edwin P. Kan

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, Ca. 91109

ABSTRACT

The Operator Control Station of the JPL/NASA Telerobot Demonstration System provides an efficient man-machine interface for the performance of telerobot tasks. Its hardware and software have been designed with high flexibility. It provides a feedback-rich interactive environment in which the Operator performs teleoperation tasks, robotic tasks, and telerobotic tasks at ease. This paper discusses the to-date operational experiences of this system, particularly related to the 'Object Designate' Process and the Voice Input/Output Process.

INTRODUCTION

The Operator Control Station (OCS) of the JPL/NASA (Jet Propulsion Laboratory / National Aeronautics and Space Administration) Telerobot Demonstration System provides the man-machine interface between the Operator and the System. It provides all the hardware and software for accepting human input for the direct and indirect (supervised) manipulation of the robot arms and tools for task execution. Hardware and software are also provided for the display and feedback of information and control data for the Operator's consumption and interaction with the task being executed. It is a large system, complex yet designed for user-friendliness.

Operational modes of the Telerobot Demonstration System include teleoperated control mode, autonomous robotic control mode, and telerobotic control mode which is a combination of the two former modes and takes the form of traded/shared control (also known as supervisory control). The execution of tasks in these various modes are all exercised from the Operator Control Station.

The OCS contains state-of-the-art mechanical and computing hardware for providing control input to the System. It contains control software and human operator interface software for real-time and user-friendly interaction. Video displays for text, graphics and camera images are provided for operator consumption. Voice input/output is provided to reduce operator work-load. Data manipulation such as 'Object Designate' capability is provided for efficient task definition and instantiation. Access to all Telerobot subsystems is provided for on-line control, monitoring, and off-line software development.

This OCS has been installed and is operational in the JPL Telerobot Testbed since April 1989. As a stand-alone system, it has offered many rewarding and satisfying hours of operational experiences in the man-machine interface, particularly in the utility of the new "object designate" technology, and in the voice input/output process. This paper will discuss these two design elements and operational aspects in detail.

OCS DESIGN REQUIREMENTS

As part of the JPL Telerobot Demonstration System, the OCS acts as the center of action for the System, and necessarily interfaces with all the other subsystems. While referring all discussions on the overall System architecture to references [1,2,3], it suffices to say that OCS interfaces with the Task Planner & Reasoning Subsystem (TPR), Run-Time Control Subsystem (RTC), Manipulator Control and Mechanization Subsystem (MCM), and the Sensing & Perception Subsystem (S&P).

The OCS is configured for two operators, the Primary Operator and the Secondary Operator (also known as the Test Conductor). The primary station of the OCS has all the controls necessary for the Primary Operator to perform all functions independent of the secondary station. And the secondary station has a subset of the capabilities at the primary station, required for monitoring and secondary control purposes. The Operators' functions include the following:

- System management functions: system startup/shutdown, setup, software configuration, other monitoring and diagnostics functions;
- System operation functions: mode transitions, setting system parameters, system calibration, video switching, emergency halt (and other modes of halting), object data manipulation;
- Manual teleoperation functions: hand motions for input to hand controllers, setting subsystem parameters, establishing telepresence via visual, kinesthetic and proprioceptive feedback;
- Autonomous robotic functions: to instantiate, monitor, supervise, direct, confirm and give permission to proceed all actions generated under autonomous planning and control;
- Telerobotic functions: to execute hybrid commands of manual and autonomous control, and transitioning between various degrees of manual and autonomous controls.
- Initiating and executing data logging functions for off-line analysis and system performance evaluation.

Further details of the OCS requirements can be found in [4]. Hardware and software design details are also documented in [5,6].

OCS HARDWARE/SOFTWARE CONFIGURATION

Figure 1 shows the OCS, as implemented at the JPL Telerobot Testbed. The Primary Operator is shown working in conjunction with the Secondary Operator in front of their respective stations. Figure 2 shows the Primary Operator looking into the robot work site area. Direct viewing of the work site is available at the JPL Testbed; and it can also be screened off so that only indirect viewing is available.

For the input of hand motions in manual telerobotic operations, the primary station houses the right and left Force Reflecting Hand Controllers and their electronics. Three overhead video monitors normally show the wing and overhead camera overviews of the task space. Two middle video monitors display special graphics required for the telerobot operations, e.g. force-torque graphics from the robot arms and hands; simulated graphics for time-delay and predicted motions; etc. Directly in front of the Primary Operator are the stereo displays, necessary for the Operator to perform dexterous close-up operations.

The primary station has three input media for command inputs into the OCS primary computer. Keyboard direct commands and system operations are provided for 'hands-off' operations, when the Operator is not using the hand controllers. A 4-tier menu input is provided for normal telerobot task operations. The same menu input commands are also duplicated by voice input, which is implemented with a flexible grammar and recognized as continuous speech commands. Iconic and hierarchical displays are also provided to the Primary Operator for the issuance of task level commands.

The secondary station has a simpler design because of its secondary requirements. It houses two video monitors and the OCS secondary computer. While the Secondary Operator does not have any voice input capability, he can always enter all the commands via the OCS secondary computer terminal, via direct command inputs using the keyboard or via the menu selection process, displayed to him on the OCS secondary computer monitor. Here, the Secondary Operator cannot provide teleoperation hand controller inputs because of the lack of the input devices. All graphics, overlays and video images can be displayed to his two monitors, as routed by an OCS process of video switching.

Common to both stations are the video switcher, which is configured to route any channels (16 input channels of RGB color) to any combination of monitors or input devices (16 output channels). E.g. single camera views can be switched to desired monitors; multi-view can be fed and mixed in a video mixer; single-view video can be fed into a graphics machine for overlaying and mixing graphics on video images. For the "object designate" process, the latter mixing is done through special graphic overlay machines routed through the video switcher.

Other audio mixing, amplification, video

encoding and decoding, recording equipments are also implemented in the OCS. A voice synthesizer is installed in conjunction with the voice recognizer. Voice annunciation is used for acknowledgement of command inputs and for the annunciation of certain messages, particularly critical messages. Additionally, both stations have their own individual emergency kill button, which can also be used for a special halt-retract function. Remote power on-off for the robot controllers are provided, and local power monitoring to the OCS racks are also installed.

The OCS computer is configured by a SUN 3/160 workstation as the primary computer, and with a SUN 3/60 workstation as the secondary computer. Both of them are connected to the ethernet for communication with the other subsystems of the Telerobot Demonstrator System. Network communication is considered more than adequate because of the low-bandwidth data rate between the OCS and the other subsystems. The multi-tasking and multi-view window capabilities on the Sun workstation are fully utilized for input and output purposes, as well as to provide terminal emulation communication to all Telerobot subsystems and users.

OCS software consists of multiple processes performing the following functions:

- Command Processing: OCS-specific commands and task level commands
- Message Processing and Display
- Ethernet and External Subsystem Interface
- Video Switch Operation and Control
- Wire Frame Object Designation
- Voice Input/Output Processing

While explanations on all these processes have been given in [6], the following discusses the last two processes in more detail.

OBJECT DESIGNATION PROCESS

This process permits the Operator to interactively update the position and orientation data of known objects by a mouse point-and-designate sequence. By so doing, any discrepancy, error, or unintentional displacement of objects - as represented in the initial data base - can by reconciled.

Task situation

Figure 3 shows the Telerobot Testbed robot work site. Two ORU (orbit replaceable unit) modules are shown sitting on the "stowbin", which is a raised and tilted rack with the appropriate SIC's (Standard Interface Connector, a standard Polar Platform design, which permits mechanical and electrical mating and coupling). The module to the right, which has a cubical shape, is better known as the electronics module; and the module to the left is the science instrument. These modules and the rack designs simulate existing designs. The SIC is a true copy of the current design (as of mid-1988).

One of the telerobot tasks to be performed at the Telerobot Testbed is to remove the electronics module from a "baseplate" location, which is shown situated between the bases of the two manipulator robots. After removal from the baseplate location (single-arm as well as dual-arm), it will be inserted into the stowbin SIC adapter.

This type of module removal, maneuver and insertion task scenario is to be performed by manual teleoperation, or by supervised telerobot control using compliant force-position control. For the latter, autonomous robot motion is commanded via 'macro' commands. And since autonomous motions are involved, 'accurate' data bases are necessary. The degree of accuracy required for the autonomous fine motions has been bounded at 5 millimeters, such that the chamfer and compliant control will permit the final correct seating.

When the stowbin or rack does not have an absolute calibrated data base entry with the degree of accuracy just stated, the technique of 'object designation' has to be invoked.

The designation process

The 'object designation' is initiated by having the following data sent to OCS: (a) object model; (b) object homogeneous transform; (c) camera models.

The object model is a wireframe description, basically consisting of a list of vertices of the object, and a list of the edges of the object. The list of vertices is accompanied by the positions of the vertices relative to a local coordinate system. The homogeneous transform of this local coordinate system origin defines the object's location and orientation, as currently stored in the data base. A bounding box can be

computed from the above data points, and the box will be used in subsequent viewing manipulations. Lastly, the geometric models of the cameras (five cameras in the current Testbed setup) are sent to the OCS; these models define the 2-dimensional coordinates on the camera focal plane when given the x-y-z location of a point in the work space.

Now the OCS is ready for the actual designation process. Up to two camera views can be selected; the two views are constrained by the two units of graphic overlay hardware in this current installation. Upon each monitor (camera view) selection, the overlaid video scene is presented on the monitor.

At the top of the monitor screen are the fourteen menu choices (to be discussed later). If indeed the wire frame of the object fits into the work space as seen by the camera's field of view, the wire frame model will appear on the screen. Otherwise, a message will appear saying that the object is out of view. In that case, the Operator can choose a 'conjure' action, forcing the presentation of the object model into the workspace. Figure 4a shows such a conjured view of the electronics module wire frame. The actual module is also visible on the right side of the video screen. Also notice the solid and dotted edges of the wire frame, providing a pseudo 3-dimensional perspective. This view is from a right wing camera.

If the module is not in a good perspective or viewing angle, certain rotations can be done to this model. The menu item at the top of screen, including 'left', 'right', 'top', 'bottom', 'rotate front' can be activated to show different views of the model.

At this point, a point selection sequence can be performed to associate vertices of the wire frame model to the vertices of the object as seen on the screen. A user-friendly clicking process has been designed into this OCS software. Figure 4b shows the state where 1 vertex has been associated (notice the circle pair surrounding the right vertex on the top face), and a second vertex is being designated (notice the 'rubber-band' cursor).

After selection of a few points on an individual screen, the point-fitting action can be activated that will exercise a least squares algorithm to fit the selected vertices to the designated locations. A minimum of three points will provide translational and rotational fitting. Any less number of points will be fitted with a translational fit only. Figure 4c shows the fitted wire frame model after three points were selected.

Usually, a single-camera-view fit will not provide a very good fit, because of the lack of 'depth'. Figure 4d illustrates the point. This is a view from a left wing camera, looking at the same fitted wire frame of Figure 4c. Even though the fit looks good in the right camera view, the need for improvement in the left camera view is evident. Normally, an orthogonal view will serve the purpose well, as in this second view. Figure 4e shows the results on the left view when an extra point, i.e a total of four points, is selected and fitted. The fit looks perfect, and so does the fit viewed from the right camera, shown in Figure 4f. After the fit, the root-mean-square error is always displayed to the Operator, on screen and by voice.

The object designation process can now be terminated by either activating 'complete' or 'cancel'. In the former, the new object location and orientation will be returned to the data base for updating.

Experiences have shown that root-mean-square errors of 1.5 pixels or less can easily be obtained consistently upon the fitting of six vertices. Six to eight vertices are normally more than enough to define an object. For shallow objects, it is not always possible to pick even that many vertices. This kind of pixel accuracy translates directly into location accuracy. Depending on the focal length and zoom of the camera, and on the accuracies of the camera models, accuracies up to 5 millimeters are attainable. For cases when such accuracies are not attained because of different sources of errors, another technique need to be invoked, namely 'relative update' technique [1].

VOICE INPUT/OUTPUT PROCESSING

The OCS has been installed with a continuous speech recognition system, the VERBEX Series 5000 system (which has its own simple voice synthesis module), and a voice synthesis system, the DECtalk DTC01 system. A custom set of vocabulary and grammar has been designed and implemented with the VERBEX system, specially designed for the OCS telerobotic operations. Standard English sentences in the form of ASCII text strings are

sent to the DECtalk for voice annunciation; and where necessary, specific phonemic phrases are sent so as to produce more natural sounding voiced messages.

Voice Input

The vocabulary and grammar set was developed according to the needs of the Operator during his operation of a typical telerobotic sequence. Essentially, this set duplicates all the keyboard or menu inputs to all the processes in the OCS software on the SUN 3/160 Primary Workstation. This set is written on the SUN workstation as a text file, called a grammar file or grammar-definition file. Using a VERBEX supplied software (for the SUN development environment), this text file is converted into a binary, machine readable recognizer file. The latter is transferred (i.e. downloaded) to the resident RAM memory and/or data cartridge in the VERBEX unit.

For the Operator to use the voice recognition system, his voice has to be trained and stored as templates on a voice file. The VERBEX supplies a user friendly development facilities for the Operator to first 'enroll' the new words in the vocabulary set, and then to train on possible combinations of phrases permissible by the grammar. Typically, a session of training for this OCS Telerobot vocabulary/grammar set is 90 minutes. Experience has shown that three training sessions normally produce very reliable voice files, where recognition accuracy and rejection accuracy could be well above 95%.

A set of 120 vocabulary words has been designed for the current OCS Telerobot operation scenarios. These words are single words as well as compound words, strung together as they are continuously spoken. Examples are: activate, cancel, clear, command_confirm, the_upper_left_display, the_stereo_cameras.

Simple grammars are designed into the use of these vocabulary words. They consist of single word commands, noun or noun phrases, and prepositions/connectors. Examples of the single word commands are: initiate, select, switch, set_video_switch_defaults, command_confirm, activate, move, verbex_report (an escape command). Examples of the noun or noun phrases are: display, camera, the_upper_right_display, the_left_wing_camera, object_designate_mode, tool_mode. Examples of the prepositions and connectors are: switch camera alpha to display charlie using channel 2.

For the two reasons of (a) that all vocabulary words are not used in one single telerobot session or operation sequence, and (b) that recognition accuracy can further be improved, the set of vocabulary words are grouped into six groups. These six groups are also consistent with the modes of telerobot operation:

- o Host mode
- o Video switch mode
- o Object designate mode
- o Vision arm manipulate mode
- o Teleoperation mode
- o Telerobot mode.

When one of these six modes are selected, a help file is also displayed on the OCS console beside the command and message windows. Figures 5(a)-5(f) show the OCS monitor displayed with the voice_command help file on the left window, during the above stated six modes. The complete voice vocabulary and grammars are listed in the help windows. (All the possible combinations can be derived from the lists, but obviously not enumerated in the window.) Shown also in the figures are the command window, normal message window and critical message window. The menu selections are also highlighted.

Voice output

In terms of voice synthesis, no special effort is needed for custom designing the voices or designing the phrases. The DECtalk provides six default voices with default parameters such as rate and pitch; where such parameters can easily be changed in the text string. The text strings sent to DECtalk need not be customized, and can be written in plain English with normal abbreviations and even with certain acronyms. Normal phrasing, with commas and periods, usually produces intelligible human speech. Where necessary to put emphasis on certian words, or where the English pronunciation deviates from standard rules, the phrase or the specific word can be written in special phonemic forms. Such phonemic forms bypass the extensive-yet-still-limited dictionary look-up rules in DECtalk, and will faithfully produce the correct accents and pronunciation so desired by the Operator or dictated by the contents and context of text phrase.

These text phrases, i.e. voice messages, are contained in a 'Message_Definition_File' which is a text file to the OCS software. Thus, the

definition of the phrases, including the specific pronunication of the phrases, could be changed at will by the Operator (i.e. user, not necessarily a programmer), without the burden of having to recompile the whole OCS software.

Likewise, since the DECtalk requires as input text files written in ASCII strings, it can be used to annunciate phrases of any language. The only limitation is the roughly 40 phonemes, i.e. basic unit of sounds, of the English language. These phonemes contain all the English vowels, unstressed vowels, dipthongs, syllabic consonants, and consonants.

SUMMARY

The JPL/NASA Telerobot Demonstration System is scheduled for complete system integration and telerobot operation in the Fall of 1989. The Operator Control Station of the System has been developed, installed, operational, and integrated into the System infrastructure. While limited amount of exposure is gained to-date on complete telerobot task executions, the Operator Control Station has offered many rewarding and satisfying hours of operational experience as a stand-alone system.

The man-machine interface has been shown to be effective, particularly in the utility of the new 'object designation' technique. Through this technique, data bases can be effectively updated and the need for absolute data base calibration is greatly reduced. Future evolution of the same technique would also bring about interactive data base construction and better graphics interface to the Operator. The voice input and output system is not just a showy convenience, but is proven to be an indispensible companion to the Operator. Even though his workload is not being reduced by these technical advances, his operational efficiency is greatly enhanced and his hands are freed to do telerobot operations. Further experience with the Telerobot System operation and the man-machine interface operation in the upcoming months will certainly suggest improvements and additions to this Operator Control Station.

Acknowledgements

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology under contract to the National Aeronautics and Space Administration. GE Aerospace/Advanced Technology Laboratory was subcontractor to JPL, responsible for the development of the Operator Control Station of the Telerobot Demonstration System.

References

[1] J. Matijevic, "The JPL Telerobotic Testbed: A System Architecture for Satellite Servicing," Proc. of 3rd USAF/NASA SOAR '89 (Space Operations - Automation and Robotics) Workshop, Houston, Tx., Jul. 1989.

[2] E. P. Kan, "System Design of a Space Telerobot System," Proc. 1988 IEEE Int. Workshop on Intelligent Robots and Systems (IROS'88), Tokyo, Japan, Oct. 1988.

[3] J. Matijevic, et.al., "Functional Requirements for the Telerobotic Testbed," Jet Propulsion Laboratory, California Institute of Technology, Document #JPL D-3693, May, 1988.

[4] E. P. Kan, "Telerobot Operator Control Station Requirements," Proc. of USAF/NASA SOAR '88 (Space Operations - Automation and Robotics) Workshop, Dayton, Oh., Jul. 1988.

[5] E. P. Kan, J. Tower, G. Hunka, G. VanSant, "The JPL Telerobot Operator Control Station: Part I - Hardware" Proc. 2nd NASA Conference on Space Telerobotics, Pasadena, Ca., Jan. 1989.

[6] E. P. Kan, B. P. Landell, S. Oxenberg, C. Morimoto, "The JPL Telerobot Operator Control Station: Part II - Software" Proc. 2nd NASA Conference on Space Telerobotics, Pasadena, Ca., Jan. 1989.

Figure 1. JPL/NASA TELEROBOT DEMONSTRATION SYSTEM - Operator Control Station



Figure 2. OCS Looking Into Robot Work Site



Figure 3. Telerobot Testbed - Robot Work Site

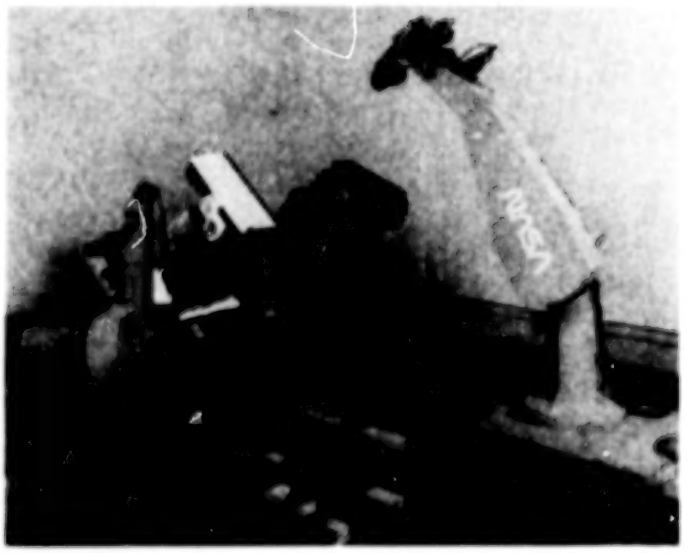
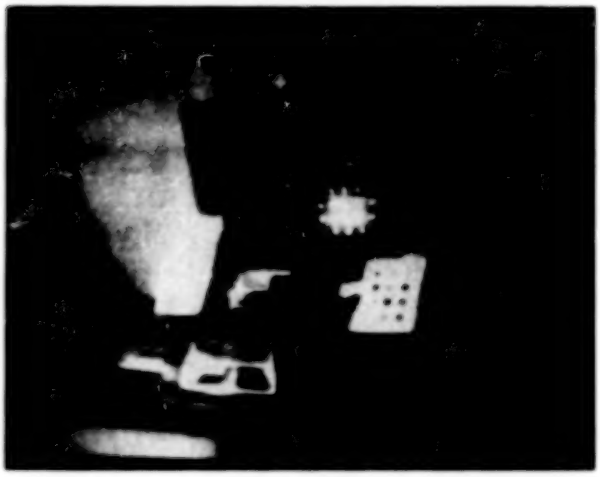
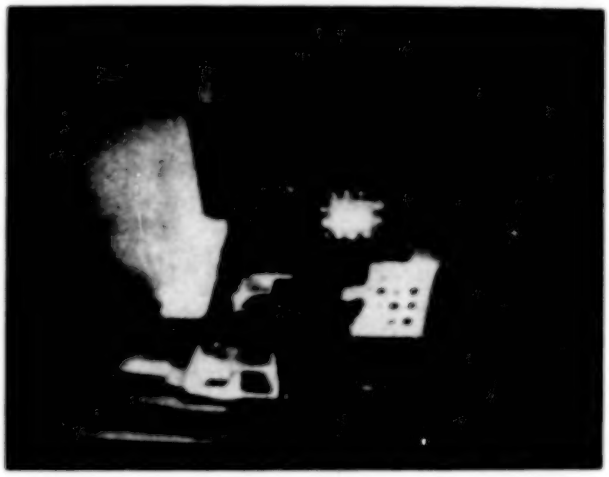


Figure 4. Object Designation Overlaid Camera Views

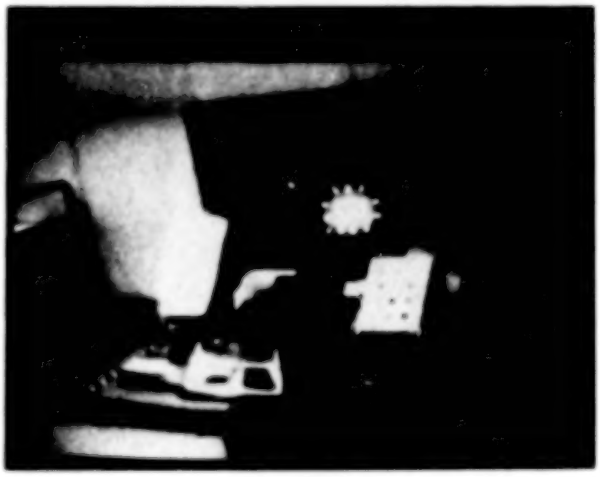
(a) CONJURED OBJECT, RIGHT VIEW



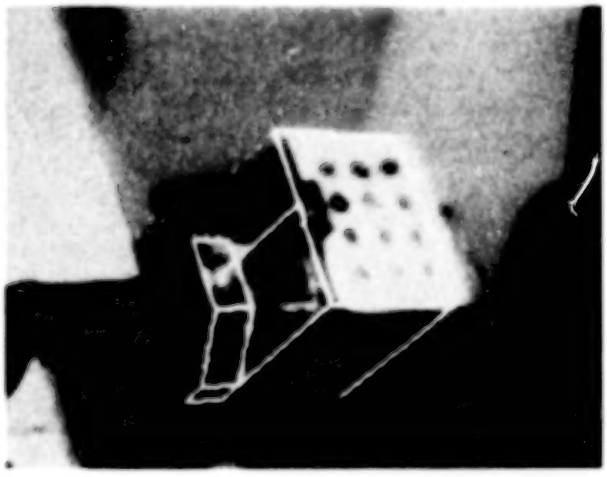
(b) SECOND VERTEX BEING DESIGNATED



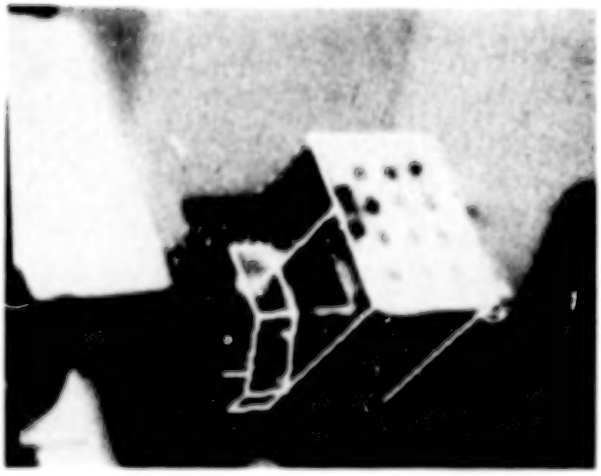
(c) FITTED IMAGE WITH 3 POINTS



(d) LEFT VIEW, USING SAME FIT AS IN (c)



(e) LEFT VIEW, 4-POINT FIT



(f) RIGHT VIEW, 4-POINT FIT

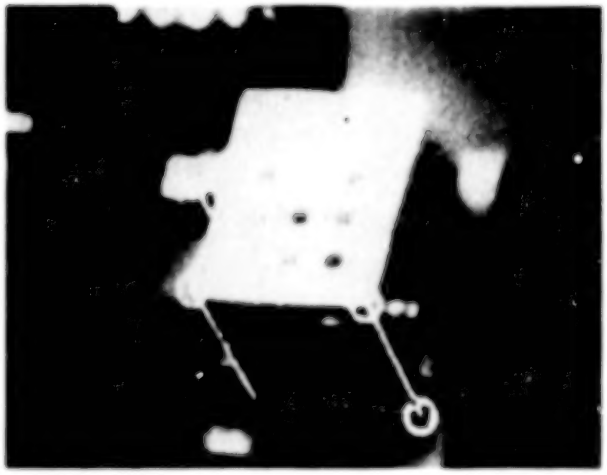
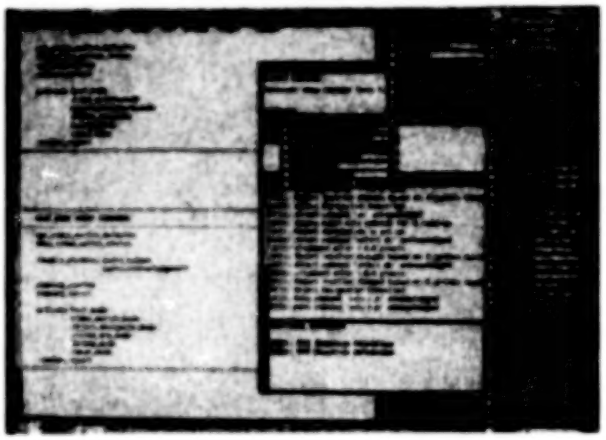
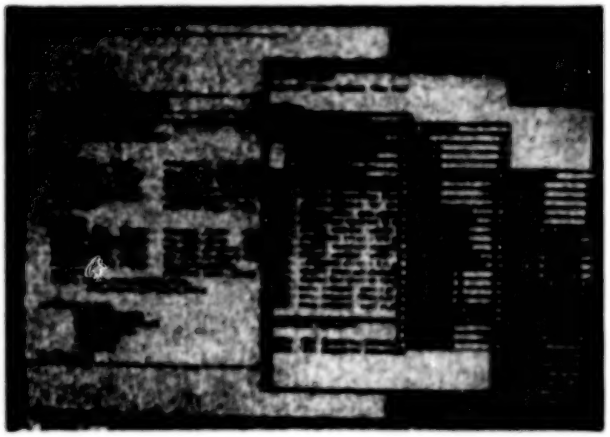


Figure 5. OCS Voice System - Grammar and Vocabulary

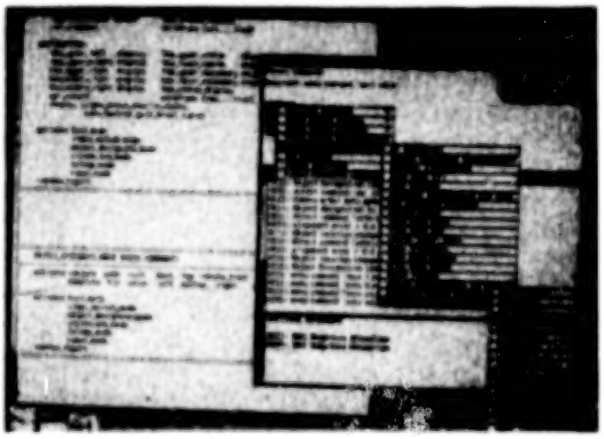
(a) HOST MODE



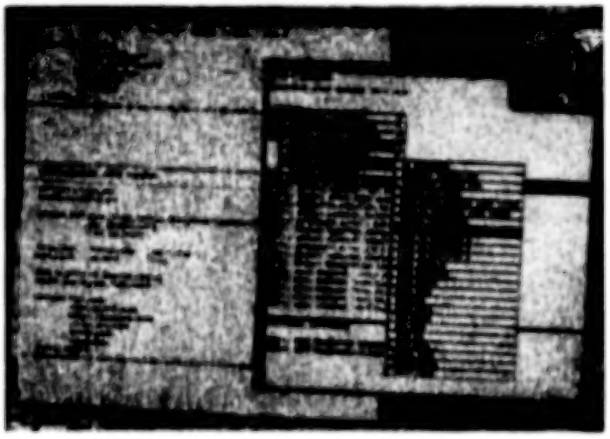
(b) VIDEO SWITCH MODE



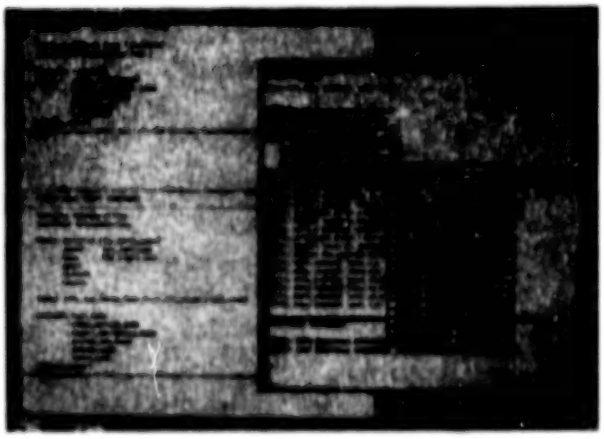
(c) OBJECT DESIGNATE MODE



(d) VISION ARM MODE



(e) TELEOPERATION MODE



(f) TELEROBOT MODE



TRAINING HIGH-PERFORMANCE TASKS WITH INTELLIGENT TUTORING SYSTEMS

Dr. J. Regian
AFHRL/IDI, Brooks AFB

(Paper not provided by publication date.)

AN INTELLIGENT SIMULATION TRAINING SYSTEM

John E. Biegel

University of Central Florida
P.O. Box 25000
Orlando, Florida 32816

ABSTRACT

The Department of Industrial Engineering at the University of Central Florida, Embry-Riddle Aeronautical University and General Electric (SCSD) have been funded by the State of Florida to build an Intelligent Simulation Training System.

One objective was and is to make the system generic except for the domain expertise. We have accomplished this objective in our prototype. The system is modularized and therefore it is easy to make any corrections, expansions or adaptations.

The funding by the state of Florida has exceeded $3 million over the past three years and through the 1990 fiscal year. UCF has expended in excess of 15 work years on the project. The project effort has been broken into three major tasks. General Electric provides the simulation. Embry-Riddle Aeronautical University provides the domain expertise. (Our first prototype demonstration is in Air Traffic Control (ATC) training). The University of Central Florida has constructed the generic part of the system which is comprised of several modules that perform the tutoring, evaluation, communication, status, etc.

This paper describes the generic parts of the Intelligent Simulation Training Systems (ISTS).

INTRODUCTION

The Intelligent Simulation Training System project is a joint effort between the University of Central Florida (UCF), Embry-Riddle Aeronautical University (ERAU) and the General Electric (GE) Company. The objective was to develop a simulation-based training system that could conduct content training without the human instructor being continuously involved.

The instructor will plan the general outline and content of a course. The system will 'intelligently' interact with the trainee during the lessons. The student will be able to communicate with the instructor and vice versa through the system. This system is extending computer-based instruction (CBI) by introducing improved interactions between the computer system and the trainee.

UCF acts as the project manager. The UCF portion of the project involves building the parts of the system (computer program code) that actually conduct the training. Care has been taken to insure that the completed system is generic so that it will be useful in many subject areas.

Embry-Riddle Aeronautical University is programming the rules by which an Air Traffic Controller actually controls air traffic. The choice of air traffic control training as a first subject area is a good choice since it will allow us to fully test our final product. ERAU has the expertise to build that set of rules.

General Electric provides the simulation through their Simulation and Control Systems Department at Daytona Beach. Simulation is an area where GE has expertise, capability, experience, and equipment.

A natural breakdown of the project is three parts:
1) simulation; 2) the rules used in the subject area; and, 3) the teaching component. The system uses the emerging technologies of computer systems

(hardware and software) and artificial intelligence (in particular expert systems technology). We have programmed the prototype on Symbolics' Machines using LISP and Flavors. We are planning to port the system to a version of the C language on a 386-based machine.

BUILDING A GENERIC INTELLIGENT SIMULATION TRAINING SYSTEM

We had many lengthy discussions on how we could build a generic ISTS. The key comes from a realization that the domains in which simulation is a viable training methodology are those in which you want the trainee to learn how to control/manipulate/understand objects in time and space. When using a simulation, the student pilot learns to control an object in time and space. The gunner/rocketeer learns how to understand or interpret what is happening in time and space to respond to it. The radar operator learns how to interpret and predict what will happen in the time-space domain. The Air Traffic Controller learns how to control objects in time and space.

In general, all simulations involve time and space and constraints on time and space. Therefore, with care, one can build a generic intelligent simulation training system in which the domain-independent and the domain-dependent knowledge are separate. We read the domain-dependent knowledge into the system at initialization.

THE SYSTEM COMPONENTS

The ISTS is programmed in modules for easier development, maintenance and for transportability. In our research of the literature relative to CBT (Computer-Based Training) and ITS (Intelligent Training Systems) we found that there were some major concerns that had not been addressed. Most of these concerns arose because of our goal of building a generic system. We found that no one had separated the evaluation system from the student model. Also, we found no systems where the domain expert and the domain instructor were treated as separate entities. We've all known domain experts who were/are poor instructors. (Also, some very good instructors would not be considered to be domain experts.) Based on our findings, we have constructed modules which we placed into six groups.

. Interface Group
  .. Author Module - a user-friendly interface for input of domain expertise.
  .. Discourse module - menu-driven communication module between the system and the user. (Input is currently through a keyboard but will be by voice at a future time.)

. Input Group
  .. Translator - accepts or rejects input on the basis of understanding: correct spelling, ability to parse and correct syntax.
  .. Input Filter - Accepts or rejects input based on the constraints of the operational domain.
  .. Intelligent preprocessor - updates current events list for the simulation.

. Control Group
  .. Interpreter - compares updated event list with pre-input event list and checks for critical events.
  .. Control - activates proper module(s).
  .. Inference Engine - makes inferences from rules.

. Instruction Group
  .. Evaluator - determines the merit of student input.
  .. Student model - maintains current status of student.
  .. Tutor - makes instructional decisions based on information from Student Model and Evaluator.

. Domain Expertise Group
  .. Domain Expert - facts, rules and heuristics for operating in the domain.
  .. Domain Expert Instructor - facts, rules and heuristics on how one instructs and evaluates in the domain.

. Simulation - appropriate simulation for instructing in the domain.

The systems modular structure is shown in Figure 1 and the data flow is shown in Figure 2. The types of data flow in Figure 2 are presented in Table 1. The Control Module oversees all of the necessary communication.

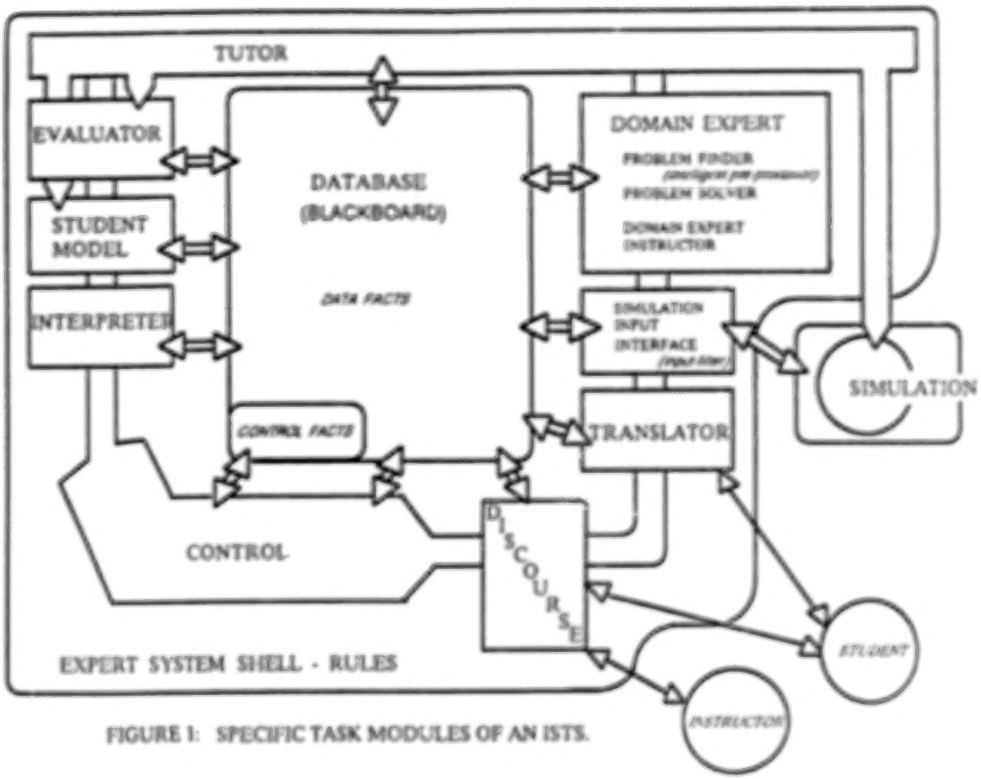


FIGURE 1: SPECIFIC TASK MODULES OF AN ISTS.

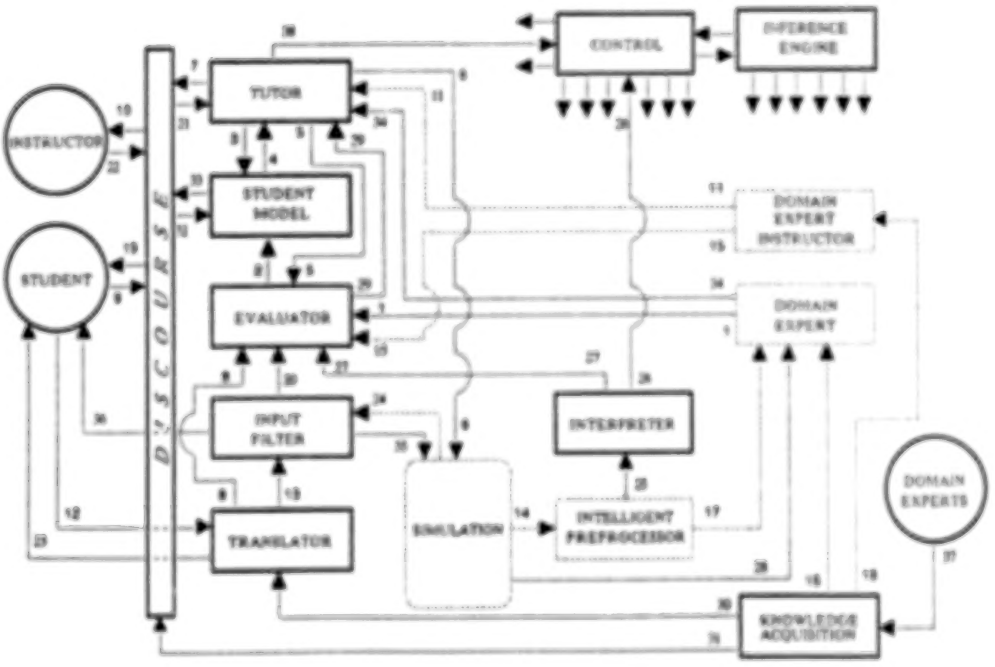


FIGURE 2: ISTS DATA FLOW DIAGRAM

Legend: ISTS modules and data flow; Interface Modules and data flow

Revised 10/03/88

Table I. Communications Within the ISTS

Key	Information Transferred
1.	Expert response to scenario.
2.	Raw score data.
3.	Refined score/status (performance rating).
4.	Remediation level and topics (to be stored).
5.	Skills/objectives to be taught.
6.	Scenario modification information.
7.	Topic, level of remediation, and final score.
8.	Parsed student commands and syntax errors (if any).
9.	All student input other than commands to simulation.
10.	Instructor inputs and requests.
11.	Domain teaching methods, lesson plans, scenarios, etc.
12.	Student response to scenario (commands).
13.	Parsed and syntactically correct student input.
14.	Plane data (vector, altitude, heading, speed), weather, emergencies, pilot requests, etc.
15.	Lesson plan, weighting factors.
16.	Domain Expert knowledge base.
17.	List of significant events (or possible events) extracted from the simulation.
18.	Domain instruction knowledge base.
19.	System's communication to student.
20.	Result's of logical analysis of student's parsed input.
21.	Student/instructor request.
22.	Communication from system to instructor.
23.	Error messages (syntax errors and uninterpretable input).
24.	Inputs refused by simulation objects.
25.	List of significant events (or possible events) extracted from the Simulation.
26.	Changes in events list.
27.	Changes in events list.
28.	Data read from simulation objects in order to generate expert solutions to events.
29.	Logical errors and specific information on student performance within the domain.
30.	Keyword and attributes for the domain.
31.	"Canned" messages for communication purposes.
32.	Student/Instructor comments.
33.	Access to student records.
34.	Why & how explanations to Domain Expert solutions.
35.	Executable message to a simulation object.
36.	Error messages sent to student (inputs refused by Input Filter).
37.	Off-line input from the system analyst.
38.	Tutor requests to activate other modules individually.

KNOWLEDGE REQUIREMENTS

An Intelligent Simulation Training System (ISTS) must contain four kinds of knowledge: (The following discussion is in reference to Figure 1.)

An ISTS must contain knowledge about the trainee. This knowledge is comprised of past performance, present performance, personal limitations and the trainee's preference as to teaching mode and type of feedback. This knowledge resides in the STUDENT MODEL.

The TUTOR contains general knowledge about teaching with simulations. The tutorial decisions about helping the student or modifying the simulation are based on dynamic information read from the EVALUATOR and the STUDENT MODEL.

The third type of knowledge is about the simulation domain. The ISTS must perceive the presence of simulation events that are of importance to a training program. This knowledge resides in the PROBLEM FINDER within the DOMAIN EXPERT Group. Since there may not be a simple solution to a particular simulation scenario, the PROBLEM SOLVER within the DOMAIN EXPERT will provide alternate solutions with a measure of merit for each. The EVALUATOR has the knowledge to evaluate the trainee's solutions by comparing them with the DOMAIN EXPERT's solutions. The ISTS must know how to monitor the syntactic and logical constraints that govern the instructions that the trainee gives to the SIMULATOR.

New knowledge about the student is gained during training sessions. The system continuously monitors for past performance (STUDENT MODEL), determines what the student has just asked the system to do (TRANSLATOR) and evaluates that action (EVALUATOR).

To use the above knowledge in a training session, it is necessary to have an interface (DISCOURSE) that allows the

instructor to set up lessons, sequence lessons, determine content, degree of difficulty, enter pass/fail criteria, etc. This module must be sufficiently versatile for the student to get help, communicate with the simulation and to comment about 'things' for the edification of the instructor.

The SIMULATOR supporting such a system must have forward data links for the passage of student actions, tutorial modifications and monitoring the actions of the simulation. It must also have feedback links for the analysis of current data and a run-time reaction speed near real-time.

Another major area is in knowledge acquisition and knowledge representation. We are working on an interface to do this in a generic way.

AIR TRAFFIC CONTROL TRAINING

Our demonstration and first prototype is being done in the Air Traffic Control (ATC) domain. This is a situation in which the trainee can and does control the moving objects in time and space. Typically, in simulation training, the student controls only one object (pilot training) or no objects (radar target detection). The control of the simulation in ATC passes from student to SIMULATOR to student, etc. That is, the student drives the simulation which then drives the student until a new decision to take action is made. After the student directs the simulation action and the simulation responds, the simulation is again driving the student.

The objective of simulation training is to train the student to visual the real-world in time and space and to understand how his/her actions alter that time-space domain.

RESULTS

We have shown that a generic intelligent simulation training system is feasible. We have also developed some concepts for such a system; namely:

- Evaluation is a process separate from the tutoring process and the student modeling process.
- There must be a domain expert knowledge base and a domain instructor knowledge base.

CONCLUSIONS

We have developed and demonstrated that it is possible to build an Intelligent Simulation Training System. Our first application has been in ATC. Any area in which simulation is a viable training/teaching method can be taught with an ISTS. To move into a new domain only requires the development of the domain-dependent knowledge which can be 'read' into the generic system.

ACKNOWLEDGEMENTS

This research was supported by the State of Florida. The work was performed by the faculty and graduate research assistants in the Intelligent Simulation Laboratory at the University of Central Florida, at Embry-Riddle Aeronautical University and at the General Electric Company (Simulation and Control Systems Department).

PUBLICATIONS AND PRESENTATIONS RELATIVE TO THE ISTS

1. Bagshaw, C.E. and Gonzalez, A.J., "Adaptive Instruction Through Intelligent Scenario Development and Modification," Proceedings of the Second Florida Artificial Intelligence Research Symposium, Orlando, FL, 1989.

2. Bagshaw, C.E., "Development of Techniques to Perform Simulation Adaptation for a Simulation Training Environment Using Expert System Methods," MSE thesis, University of Central Florida, 1988.

3. Biegel, J.E. and co-author, "The Use of Expert Systems to Develop an Intelligent Simulation Training System," IEEE Conference on Management and Technology Proceedings, 1987.

4. Biegel, J.E., "Intelligent Training Systems," IIE Integrated Systems Conference, 1987.

5. Biegel, J.E. and co-authors, "Control and Interfacing Paradigms for an Intelligent Simulation Training System, Proceedings of the First Florida Artificial Intelligence Research Symposium, Orlando, FL, May 1988.

6. Biegel, J.E. and co-authors, "Input and Instruction Paradigms for an Intelligent Simulation System," Proceedings of the First Florida Artificial Intelligence Research Symposium, Orlando, FL, May 1988.

7. Biegel, J.E. and co-authors, "Student Modeling in an Intelligent Simulation-Based Training System," Proceedings of the 1988 Southeastern Simulation Conference, Orlando, FL, October 1988.

8. Biegel, J.E., "The Essential Components of an Intelligent Simulation Training System," Proceedings of the 1989 SCS Western Multiconference, San Diego, CA, 1989.

9. Biegel, J.E. and co-author, "Inferencing in an Intelligent Simulation-Based Training System," Proceedings of the 1988 Southeastern Simulation Conference, Orlando FL, October 1988.

10. Dixon, C., "A Student Performance Evaluation Method for an Intelligent Simulation-Based Tutoring System," MSE thesis, University of Central Florida, 1988.

11. Gonzalez, A.J. and co-authors, "A Simulation-Based Expert Systems for Training Air Traffic Controllers," Proceedings of the First Florida Artificial Intelligence Research Symposium, Orlando, FL, 1988.

12. Interrante, L.D. and Biegel, J.E., "The Role of Knowledge Representation in Knowledge-Based System Design," Proceedings of the Second Florida Artificial Intelligence Research Symposium, Orlando, FL, 1989.

13. Interrante, L.D. and Biegel, J.E., "Automatic Knowledge Acquisition for an Intelligent Simulation Training System," Proceedings of the 1989 Conference of the International Association of Knowledge Engineers, Washington, D.C., 1989.

14. Lee, C.H. and co-authors, "Student Performance Evaluation for a Simulation Based Intelligent Expert Tutoring System," Proceedings, Human Factors Society 32nd Annual Meeting, Anaheim, CA, October 1988.

15. Roney, A., "A Frame-Based Knowledge Representation Scheme for an Expert System," MS research report, University of Central Florida.

16. Sargeant, J., "A Dynamic Performance-Based Student Model for an Intelligent Simulation Training System," MSE thesis, University of Central Florida, 1988.

FUNCTIONAL DESCRIPTION OF A COMMAND AND CONTROL LANGUAGE TUTOR

David R. Elke and Thomas L. Seamster
Carlow Associates Incorporated
8315 Lee Highway
Fairfax, Virginia 22031

Walter Truszkowski
Code 522.3
Goddard Space Flight Center
Greenbelt, Maryland 20771

ABSTRACT

This paper describes the status of an ongoing project to explore the application of Intelligent Tutoring System (ITS) technology to NASA command and control languages. The primary objective of the current phase of the project is to develop a user interface for an ITS to assist NASA control center personnel in learning Systems Test and Operations Language (STOL). Although this ITS will be developed for Gamma Ray Observatory operators, it will be designed with sufficient flexibility so that its modules may serve as an ITS for other control languages such as the User Interface Language (UIL). The focus of this phase is to develop at least one other form of STOL representation to complement the operational STOL interface. Such an alternative representation would be adaptively employed during the tutoring session to facilitate the learning process. This is a key feature of this ITS which distinguishes it from a simulator that is only capable of representing the operational environment.

INTRODUCTION

This paper describes the status of an ongoing project to explore the application of Intelligent Tutoring System (ITS) technology to NASA control centers. The primary objective of the current phase of the project is to develop a user interface for an ITS to assist NASA control center personnel in learning Systems Test and Operations Language (STOL) with the aim of designing the ITS with sufficient flexibility so that its modules may serve as an ITS for other control languages such as the User Interface Language (UIL).

The paper first addresses nine broad areas of functionality that combine to produce an ITS. This presentation emphasizes that these functions may have different levels of implementation, from a very simple level to a complex one requiring considerable computational resources. This approach decomposes the ITS into functions that do not match the traditional ITS modules (see Figure 1 for the relationship between the functions and the ITS modules). The reason for this decomposition is to take a fresh look at ITSs from the perspective of NASA command language training needs. The nine functions are as follows:

1) *Initiating* the tutoring session
2) *Assessing* the student's status
3) *Presenting* the problem
4) *Monitoring* the student's performance
5) *Assessing* the student's goal
6) *Identify* the information to be tutored
7) *Adapting* tutor mode to student
8) *Tutoring* the student
9) *Updating* the student model

The last part of the paper presents the critical issues affecting the current phase. These issues include which modules will be developed first, which functions will be given the highest priority in the ITS, and the process for deciding on whether to use an intermediate form of representation for the tutoring of the control language and/or the objects being controlled.

INITIATING THE TUTORING SESSION

An ITS may initiate a tutoring session in response to one or more of the following events: 1) a request from the student for instruction; 2) a pre-defined schedule of tutoring; or, 3) detection by the ITS of a flaw in the

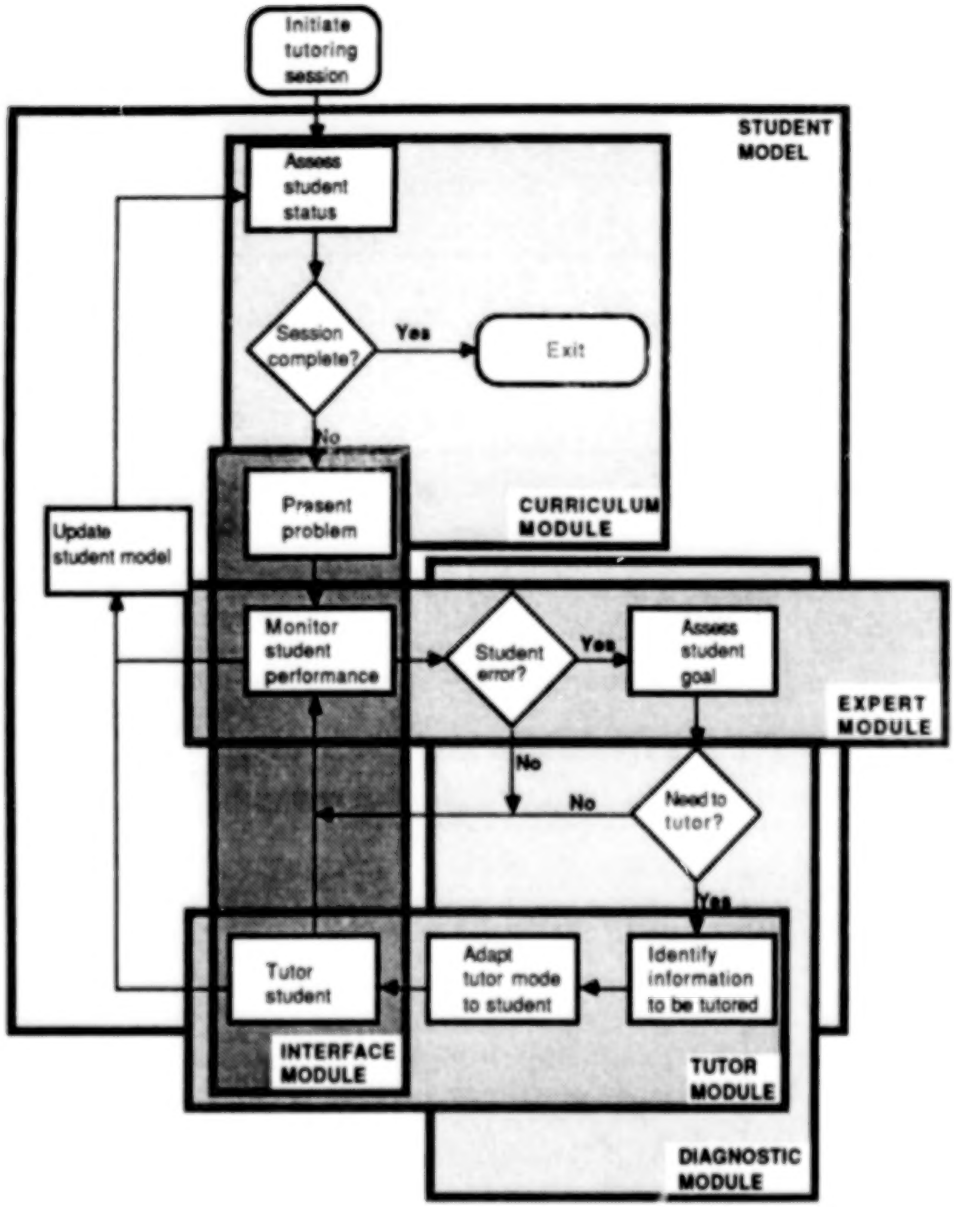


Figure 1. Functional Flow of a Command Language ITS

student's knowledge. Each of these approaches has significant implications for the design and overall functional characteristics of the ITS.

The simplest ITS architecture employs the programming equivalent of an "on" switch to initiate a tutoring session. In this approach, the student recognizes the need for tutoring and requests that the ITS begin tutoring. At a more complex level, the ITS may have the capability to infer what instruction is necessary by examining the student model.

An increasingly complex ITS would include the use of a tutoring schedule which describes the timing and sequencing of individual "lessons." The schedule is developed as part of the ITS's curriculum module, and modified, as necessary, to accommodate requirements of the individual student. The ITS queries the student model to establish the student's position within the schedule, and then initiates the appropriate tutoring session to advance the student to the next scheduled level of learning.

ASSESSING STUDENT'S STATUS

The ITS must infer the student's status within the session as well as the entire tutorial. Depending on the modules of an ITS, this assessment may be based on the interaction of the data in the student model with that of the curriculum module. At the start of a session, this assessment will be instrumental in determining which problem to present. This assessment serves also to determine if the session is complete.

PRESENTING THE PROBLEM

In its simplest form, this function involves selecting the next problem from a predetermined problem set. The assumption here is that the ITS's primary function is to present domain related problems to the student, and to provide tutoring when the student has difficulty with the problem. A more sophisticated implementation of this function would take into account a number of factors in the student model in order to more closely tailor the problem selection and representation process to the individual student. Such elements as previous errors, difficulties with specific concepts, and the recency with which the student has used the tutor could combine to guide the optimum problem selection.

MONITORING THE STUDENT'S PERFORMANCE

This function requires that there be a common workspace for the ITS and student, and that the ITS is capable of monitoring the student's performance in some meaningful context. At least five levels of student monitoring are possible: keystroke, word, phrase, consequence, and complete student product.

There are obvious advantages and disadvantages for each of the above levels. Burns and Capps (1988) refer to this as the "bandwidth question." At the level of individual keystrokes, the ITS may achieve the greatest precision. The cost of this precision is registered in terms of development effort and potential for "abuse." In terms of development effort, the cost is associated with developing models of the user and the system which can be meaningfully examined at the level of individual keystrokes. In terms of "abuse," the keystroke level of monitoring provides the greatest opportunity for intrusive intervention, what might be termed the "backseat driver" effect. As the level of monitoring increases in granularity, the advantages and disadvantages shift accordingly, until at the total product level, the ITS may be too imprecise to be effective, but is virtually incapable of intrusive intervention.

ASSESSING STUDENT'S GOAL

If the student's performance deviates from that of the expert module, or is in error for some other reason, the ITS must assess the student's goal for the current transaction. In the present context, this will be necessary to evaluate the adequacy of the student's semantic knowledge of STOL. At a minimum, the ITS should be able to infer the student's high-level goal based on the current problem.

A more accurate definition of the student's goal may be achieved by incorporating data on the student's mental models of the current problem domain. The ITS may generate an hypothesis about how the student conceptualizes the current problem, both as an individual entity and as part of the problem domain. This hypothesis should describe both the student's problem conceptualizations as well as problem solving strategies.

The student model traditionally supports this function by providing two types of information. First, the student model provides a general description of how the typical student conceives of and solves specific problems in the problem domain. Second, the student model contains an historical record of the current student, including aptitudes, the content and outcome of previous tutor sessions, typical errors, and preferences for tutor session design and content. This information in used to infer the student's current goal and to formulate preliminary hypotheses about the student's current level of knowledge.

Given a student goal, the ITS can determine if tutoring is required. Not every student error will require tutoring. For example, an ITS may be capable of distinguishing between errors and what Carroll and McKendree (1987) refer to as "characteristic nonoptimalities" (i.e., idiosyncratic problem solving strategies which are effective but not optimal). For this capability, the ITS must be sensitive to individual problem solving strategies which may range outside the normative model, but which are effective and consistent with both the both student's objectives and his conceptualization of the problem. The concern here is not only with regard to the student's perception of the acceptability of the ITS, but also in terms of maximizing the student's natural propensity to learn. Woolf and Cunningham (1987) recommend developing an ITS which can guide a students without interruption as long as the student appears to be attaining a specific goal.

SELECTING THE INFORMATION TO BE TUTORED

At a minimum, the ITS selects information to be imparted to the student based on explicit errors which are indicative of a specific flaw in the student's knowledge. At a practical level, the ITS can infer knowledge gaps by comparing the student's performance to that of the expert using a normative model which reflects changes in student behavior as they acquire skills and knowledge. In general, this approach assumes that the knowledge gaps of the student can be inferred once the student's position on a hypothetical skill acquisition continuum has been defined. This approach is implemented through the development of a general model of student performance which describes changes in the way the student solves problems at various stages of expertise. Given such a model, we can elaborate on individual stages using system specific examples. A major limitation of this approach is that it must assume that the knowledge level of the student is monolithic for a particular stage of learning. This limitation may be at least partially overcome by including independent student models for each relevant knowledge domain. This solution, however, raises significant implementation issues regarding efficiency and economy.

Wenger (1987) proposes a different solution to this problem which involves development of a diagnostic model of the student's current skills and subskills. This model is based on a representational scheme for procedural skills with an emphasis on those skills that can be mislearned. The notion of evaluating individual student actions, in terms of the implications of those actions for the skill/knowledge level of the student, may be thought of as an assessment of the ignorance of the student. The term, ignorance, is used here in its clinical sense without pejorative connotation. What is required, then, is a descriptive and computational model which can be used to evaluate the type and amount of ignorance an individual possesses with regard to a particular element of the knowledge domain. This shifts the focus of the ITS's assessment of the student from what the student knows to what the student does not know. This assessment may more directly support the selection of knowledge to be tutored than does the development of traditional student models. This requirements for assessing ignorance are discussed in detail in Elke and Seamster (1989).

There are at least 4 distinct states of ignorance which must be considered:

1. Unknown/misknown class of elements - the student is not aware of the existence of an entire class of elements or has mistaken notions about the class. In this case, the ITS will have to explain to the student what are the characteristics which define membership in this class.
2. Unknown/misknown element - the student is not aware of the existence of the element or does not believe that the element is a member of a particular class. The student has a valid schema for the class of elements to which the current item belongs, but is unaware of the existence of this particular instance of that class or has not assigned the element to the proper class.
3. Unknown/misknown relevance - the student is aware of the existence of the object but has failed to observe the relevance of the object for the present problem. The basis for this ignorance may be more profound, necessarily deriving from a failure to recognize the relationship between known attributes of known objects.
4. Unknown/misknown rules or procedures - the student is aware of all of the relevant objects and classes of objects, but lacks or misunderstands the rules necessary to solve the problem.

In order to instantiate this approach, the ITS needs the capability to detect an error and then infer the underlying ignorance. In selecting the information to be imparted, the ITS may consider several information parameters, including content, amount, level of detail, and format. As shown in Figure 1, the diagnostic module interacts with the student model and the curriculum module in making this selection.

ADAPTING TUTORING MODE TO STUDENT

A major distinction between conventional Computer Assisted Instruction (CAI) and an ITS is the ability to dynamically adapt the tutoring strategy to the current needs of the student. In theory, one of the most powerful features of an ITS is the ability to alter its mode of instruction to accommodate the unique requirements of the individual student. In practice, however, this capability may be beyond the current state of the art. Carroll & McKendree (1987) observe that current ITSs are not able to reason about tutoring or select strategies dynamically.

Wenger (1987) describes a number of potential tutoring strategies, including the following:

Case method
Coaching
Engage and pull
Issues and examples
Model tracing
Modeling-scaffolding-fading
Planning nets
Steering testing
Socratic method

The current concern is in developing a partially adaptable ITS. In the case of a STOL ITS, the main issue it whether the ITS should have more than one tutoring strategy, and how would several strategies be selected and presented to the student. Norcio and Stanley (1988) discuss several negative aspects of adaptive interfaces which have implications for ITS design. First, adaptation, by definition, involves a change in the way the ITS interacts with the student. According to Norcio and Stanley, such changes may inhibit the user's ability to develop a coherent model of the system. This has the potential effect of undermining the student's confidence in his understanding of the system with a consequent degrading of the student's performance. Similarly, the student may experience a sense of losing control over the system such as not being able to predict the system's response. This also may contribute to a general feeling of confusion on the part of the student. From a practical perspective, adaptation imposes significant development costs.

TUTORING THE STUDENT

Given that the ITS has identified and diagnosed the student's problem, and developed a general plan for remedial action, the system is ready to begin advising or tutoring the student. This function contains those elements of the ITS that involve the actual communication of information to the student. Irrespective of the tutoring strategy employed by the ITS, the system should have the ability to explain the rules and information processes which underlay its knowledge base. This feature is typically referred to as the "glass box model" (Burns and Capps, 1988), and is similar to "explanation facility" used in expert systems.

In order to be maximally useful to the student, the ITS should have the following capabilities relative to explaining the rules contained in its knowledge base:

1) At any point during a tutoring session, the ITS should be capable of displaying the rules which are central to solving the current problem.
2) The ITS should be capable of recalling and displaying each invoked rule and associating it with a specific event to explain the rationale for the ITS's assessment of the event.
3) The tutor module should be able to search the knowledge base to locate rules or items of knowledge in response to specific inquiries from the student.

The ITS should provide the capability to model and predict the performance of an expert in solving a particular problem. In this manner, the student could observe the expert's problem solving strategies in a context that is relevant to the student. The tutor module would then be able to "play back" the expert's solution, step by step, with the student examining each step and querying the ITS for explanations and justifications.

As indicated in Figure 1, the tutoring function involves the tutor module, the student model, and the user interface module.

UPDATING THE STUDENT MODEL

The final function of the ITS is to update the student model. The student model should be updated based on the results of the activities which occurred during the session. Figure 1, depicts the evaluation function as being driven by the output of the tutor module in conjunction with the data on the student's performance. This is a somewhat simplified version of the update function. For a more complex version, information collected or generated by most of the remaining modules throughout the session could be incorporated into the update. This more complex form of updating could pose difficult data management problems.

CRITICAL ISSUES

During this phase of ITS design, there are a number of critical decision that are being made affecting the direction and final capability of the STOL ITS. A major decision is to emphasize the user interface. A number of the earlier ITSs initiated with the development of an expert module. This is due in part to the fact that some of these ITSs evolved from expert systems and consequently had established expert modules. In the case of STOL, there is not an existing expert system, and it was decided to start off with the development of whichever module would provide the best data for evaluating the feasibility of the project.

ITS development has matured to the point that greater emphasis can now be placed on the end-user as well as the ITS's impact on the training system. Given this new focus on the end-user, it was decided that the user interface module would provide a good starting point. A user interface prototype could be used to gather the traditional interface preference and performance data, and could be used to evaluate the relative effectiveness of alternative tutoring strategies in the context of specific student problems.

In discussing the development process of existing ITSs, it was discovered that some of the current ITSs were developed starting with the interface. Specifically, the Geometry Tutor (Anderson, Boyle, & Yost, 1985) was developed with the interface being completed first. One of the reasons for this is that the user interface for the Geometry Tutor is relatively complex, and the representation of the geometry proofs was considered to be a critical factor in the success of the Geometry Tutor. Analysis of this development approach revealed an additional benefit from starting with the user interface. If the user interface prototype were sufficiently flexible and robust, that prototype could be used to not only gather user data, but could be used as the primary tool for knowledge acquisition. STOL experts would interact with several sample problems and the various forms of representing STOL, and would provide a range of problem solutions as well as ways of optimizing STOL representations.

This early emphasis on the user interface has influenced the emphasis that will be placed on the nine functions of an ITS. Table I shows those relative emphases.

Table I. ITS Functions and Their Relative Emphasis

FUNCTION	FOCUS OF THIS PHASE	LEVEL OF IMPLEMENTATION	
		Simple	Complex
Initiating the tutoring session		•	
Assessing the student's status			•
Presenting the problem	•		•
Monitoring the student's performance		•	
Assessing the student's goal		•	
Identify the information to be tutored	•		•
Adapting tutor mode to student	•		•
Tutoring the student	•		•
Updating the student model		•	

The STOL ITS will represent STOL as a command language in its linear textual form in the context of the STOL interface as shown in Figure 2. In addition, the designers would like to develop at least one other form of STOL representation that would facilitate the learning and tutoring process. This would be a key feature of the ITS that would distinguish it from a simulator, just capable of representing the operational environment. The psychological motivation for this is that student's of programming languages can easily overload their working memory with syntax rules. One way to reduce this cognitive workload, is to provide the student with a structured editor such as the one in the LISP Tutor (Anderson & Reiser, 1985). With that editor, when the student enters a LISP function, the tutor displays place holders for the required arguments. The editor also automatically balances parentheses. This form of editor does reduce cognitive workload, but it fails to solve another problem common among programming language students, their difficulty in translating their natural language solutions into the narrow and restrictive environment of most programming languages. One solutions to this problem, proposed by Bonar et al. (1988), is to provide the student with some intermediate representation of the programming language that will be less taxing to the student's working memory and at the same time provides an easier transition from the normal way of structuring the problem.

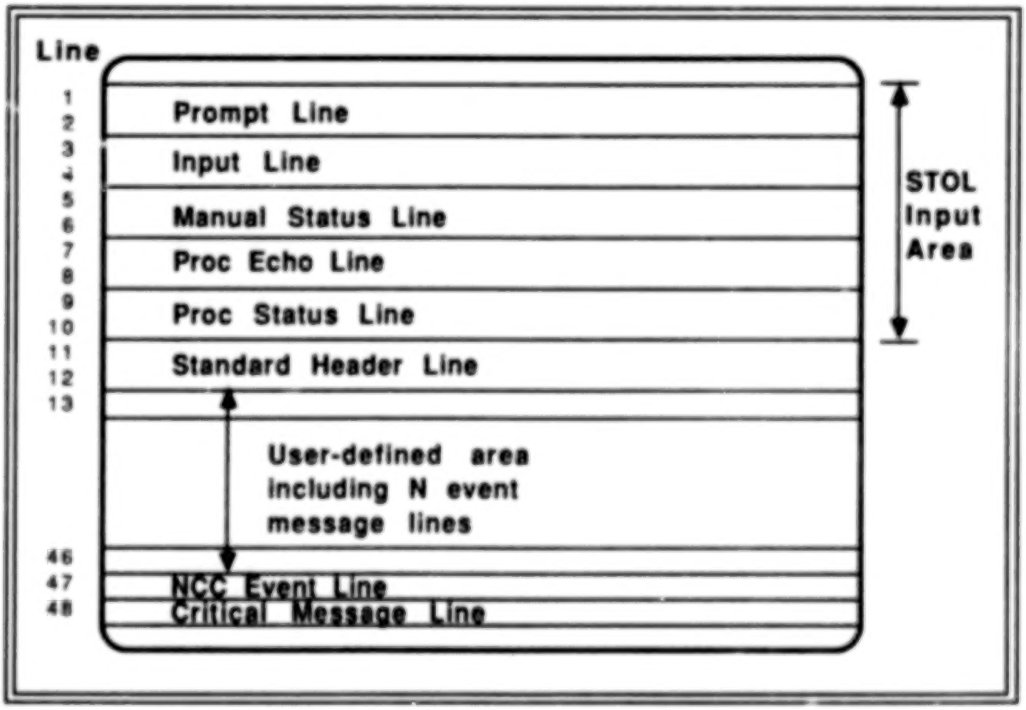


Figure 2. The STOL User Interface

In the case of Bonar et al. (1988), this intermediate representation took the form of a visual programming language. More precisely, they used a set of icons to represent the programming plans required to solve a limited set of Pascal programming problems. These icons were in the form of puzzle pieces emphasizing the placement of these plans in relation to other plans. A similar approach may be helpful for STOL students, and a several forms of visual programming languages and direct manipulation environments will be evaluated.

STOL was selected as the command language for this phase of the ITS implementation. STOL is used in a number of Payload Operations Control Centers (POCCs) to control ground system elements. It allows for the control of telemetry, command and display processing, and related support functions. There were a several key considerations in selecting a specific POCC. First, the POCC had to have a number of active controllers using STOL that would serve both as subject matter experts and as evaluators of the STOL ITS prototype. The POCC had to have a number of STOL controlled tasks that were similar to the majority of other POCCs, and the controllers had to be active and accessible during this phase of the ITS design.

The Gamma Ray Observatory (GRO) POCC met the above criteria and was selected for this phase of the ITS development. There are currently six to eight GRO operators, with that number increasing to 12 or 14 during critical activities such as the launch. These operators will be available to provide both the preliminary STOL data and the in-depth GRO task information. Once the prototype is developed, these operators will be able to provide problem solutions as well as ways of optimizing STOL representations.

The goal of the STOL tutor is to provide the critical training to controllers who need to learn the language. For the early stages of development, the STOL ITS will be limited to STOL as it is used in the GRO POCC, and as it is applied to a limited range of payload control problems. This is a similar approach as was taken in the development of PROUST (Johnson, 1986), a tutor that analyzes Pascal programs for non-syntactic bugs. During the development phase, PROUST was designed to analyze two programming problems. The developers demonstrated PROUST's ability to diagnose hundreds of novice solutions to these two problems. They wanted to establish PROUST's diagnostic robustness before expanding the number of problems that it could diagnose. During this phase of the ITS design, several GRO POCC problems will be identified for the tutoring process. These problems should be similar to problems solved through STOL at other POCCs. They should be problems which are relatively difficult for novices to solve and that require a range of STOL skills. Finally, these problems should be related to tasks that are critical to mission success.

The ultimate goal of this project is to develop a set of ITS modules with sufficient flexibility so they may be used for the tutoring of other command languages used at NASA control centers.

ACKNOWLEDGEMENTS

Financial support for this endeavor has been provided by CODE 522.3 of Goddard Space Flight Center. We would also gratefully acknowledge the help of Henry Murray, CODE 511.2 of Goddard Space Flight Center, in coordinating interviews with subject matter experts in the GRO POCC.

REFERENCES

Anderson, J. R. (1988). The expert module. In M. C. Polson, & J. J. Richardson (Eds.), *Foundations of intelligent tutoring systems* (pp. 21- 53). Hillsdale, NJ: Lawrence Erlbaum Associates Publishers.

Anderson, J. R., Boyle, C. F., & Yost, G. (1985). The Geometry Tutor. *Proceedings of the International Joint Conference on Artificial Intelligence*. Los Altos, CA: Morgan Kaufmann Publishers, Inc.

Anderson, J. R., & Reiser, B. J. (1985). The LISP tutor. *BYTE*, 10:4, 159-175.

Bonar, J., & Cunningham, R. (1988). BRIDGE: An intelligent tutor for thinking about programming. In J. Self, (Ed.) *Artificial Intelligence and Human Learning* (pp. 391-409). New York: Chapman & Hall Ltd.

Burns, H. L. and Capps, C. G. (1988) Foundations of Intelligent Tutoring Systems: An Introduction in *Foundations of Intelligent Tutoring Systems* Polson, M. C., and Richardson, J., (eds) Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Carroll, J. M., & McKendree, J. (1987). Interface design issues for advice-giving expert systems. *Communications of the ACM*, 30, 14-31.

Eike, D. & Seamster, T. L. (1989). Application of ITS technology to NASA control centers. Final Technical Report under Contract SEAS CAR-2911800.

Johnson, W. L. (1986). Intention-based diagnosis of novice programming errors. Los Altos: Morgan Kaufmann Publishers, Inc.

Norcio, A., Stanley, J. (1988). *Adaptive Human Computer Interfaces*. NRL Report 9148, Washington, D.C.: Naval Research Laboratory.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. Morgan Kaufmann Publishers, Inc.

Woolf, B. (1988). 20 years in the trenches: What have we learned? *Proceedings of the International Conference on Intelligent Tutoring Systems*.

SUCCESS IN TUTORING ELECTRONIC TROUBLESHOOTING

Ellen M. Parker
Air Force Human Resources Laboratory
Manpower and Personnel Division
Brooks AFB, TX 78235-5601

ABSTRACT

Two years ago at this conference, Dr. Sherrie Gott of the Air Force Human Resources Laboratory described an avionics troubleshooting tutor being developed under the the Basic Job Skills Research Program. The tutor, known as Sherlock, is directed at teaching the diagnostic procedures necessary to investigate complex test equipment used to maintain F-15 fighter aircraft. Since Dr. Gott's presentation in 1987, the tutor has undergone field testing at two Air Force F-15 flying wings. The results of the field test showed that after an average of 20 hours on the tutor, the 16 airmen in the experimental group (who averaged 28 months of experience) showed significant performance gains when compared to a control group (having a mean experience level of 37 months) who continued participating in the existing on-the-job training program. Troubleshooting performance of the tutored group approached the level of proficiency of highly experienced airmen (averaging approximately 114 months of experience), and these performance gains were confirmed in delayed testing six months following the intervention. The tutor is currently undergoing a hardware and software conversion from a Xerox Lisp environment to a PC-based environment using an object-oriented programming language. This paper summarizes the results of the successful field test and focusses on (a) the instructional features that contributed to Sherlock's success, and (b) the implementation of these features in the PC-based version of the avionics troubleshooting tutor.

INTRODUCTION

In developing the avionics troubleshooting tutor to be described in this paper, the Basic Job Skills Program attempted to address several fundamental problems that the Air Force maintenance community faces with respect to the training of maintenance technicians. First, while the complexity of the systems to be maintained is increasing with advances in aerospace technology, there has been no corresponding increase in the time available to new trainees to learn about these systems. As a result, the time needed to acquire the knowledge necessary to perform these jobs increases and the Air Force has received fewer of the benefits of its training by the time maintenance personnel leave the Air Force.

One response to the increase in the technical complexity of these jobs has been to provide technicicans with proceduralized job aids and so-called 'smart' machines equipped with self-diagnostic capabilities. The rationale supporting this response is that in providing technicians with cook-book procedures for dealing with maintenance problems they might encounter, together with machines that diagnose their own faults, one can reduce the knowledge and skill required of the human technician and still maintain the productivity of the work force. Unfortunately, the adequacy of proceduralized job aids is limited by the fact that, given the complexity of current aircraft systems, even the best designers are unable to anticipate every conceivable fault or fault combination that their system might develop. There are also limits to the diagnostic capabilities of automated systems. For example, Gott (1987) cited a 65% hit rate for the diagnostics of some systems on the B1B. Thus, there is still a clear need for human expertise to pick up where procedural aids and automated diagnostics leave off. A related consequence is that reliance on such aids gives technicians a false sense of security and undermines the development of the expertise that will invariably be required when the technician is confronted with a novel set of conditions for which the proper repair procedures have not been prespecified.

A final dimension of the maintenance training has to do with the fact that the

first priority in the shops where technicians receive their on-the-job training is rarely to train new technicians, but to keep planes flying. This means turning out units that come into the shop for repair as quickly as possible. In order to meet this goal and to keep inexperienced personnel from breaking expensive equipment, it is often only the most experienced technicians who work on the more difficult problems. Thus, trainees are denied important learning opportunities where practice at solving difficult diagnostic problems would promote their understanding of the task and the system they are working with.

DEVELOPMENT AND EVALUATION OF AN AVIONICS TROUBLESHOOTING TUTOR

In order to ameliorate these effects, the avionics troubleshooting tutor, Sherlock, was designed to provide trainees with the type of troubleshooting practice that would decrease both the total reliance of novice technicians on automated diagnostics and proceduralized job aids, as well as the amount of time required to acheive proficiency in the task of maintaining aircraft systems. The design was based on analyses of expert troubleshooting performance (Gitomer, 1984, 1988; Glaser et al., 1985; Gott, Bennett, and Gillett, 1986) which identified three cognitive components of their expertise: the knowledge underlying experts' use of troubleshooting procedures such as tracing electrical signals using schematics and taking measurements of the signals; the strategic knowledge underlying decisions regarding appropriate actions to take given multiple alternatives; and the declarative knowledge of the system itself which allows experts to accurately represent the problem and thereby construct and constrain the problem space. Sherlock incorporates a series of 34 troubleshooting scenarios that are designed to foster these multiple types of expert knowledge. The scenarios are presented to students in an ordered sequence. This sequence was informed by the examination of novice weaknesses in the cognitive task analysis, and was designed to foster increasingly sophisticated models of the test equipment and the troubleshooting task.

Sherlock was evaluated in a controlled experiment at two Air Force F-15 flying wings (Nichols, Pokorny, Jones, Gott, and Alley, in preparation; Gott, 1989). A verbal troubleshooting test was used to identify 32 avionics technicians who had either beginning or intermediate troubleshooting skills (see Nichols, et al. for a complete description of the verbal troubleshooting task). On the basis of their performance, subjects were ranked within testing site and matched pairs were established. One member of each matched pair was then randomly assigned to either the experimental or control group such that half the subjects at each testing site were assigned to each group. Subjects' scores on the verbal troubleshooting task provided a baseline measure against which performance gains could be measured post-experimentally. The pretest scores revealed no significant differences between groups in performance on either the verbal troubleshooting problems that were administered at both bases or on a number of other indicators that were used to corroborate the equality of groups prior to the intervention (see Nichols, et al. for a complete description of these measures).

The experimental subjects received an average of 20 hours on Sherlock over the course of approximately three weeks while the control subjects continued their on-the-job training. Parallel forms of pretest measures were then readministered as posttests by researchers who were blind with respect to individual subjects' participation in either the experimental or control group. Figure 1 shows differences in pre- and post-test performance on the verbal troubleshooting task for the two groups. An independent samples t-test revealed no significant differences between mean pre-test scores of 53.40 for the control group and 56.93 for the tutor group ($t(30)=0.38$, $p>.5$, two tailed). Post-test performance, however, differed significantly ($F(1,29)=15.62$, $p<.01$), with tutored subjects obtaining a mean score approximately 20 points higher than that of control subjects. In order to get some idea of what this performance gain translates to in terms of increased experience, a group of skilled airmen with an average of 114 months of experience in this career field was tested on the verbal troubleshooting task. Their mean score is plotted in the upper left-hand corner of Figure 1, and is quite similar to that of the tutored group who had an average of only 28 months of experience. When experimental and control subjects were retested 5 to 6 months after the experiment had been conducted, the tutor's effect persisted with tutored subjects achieving a mean score approximately 15 points higher than that of the control group. When compared with their immediate posttest performance, the slight performance decrement of the tutored group on the delayed posttest was not statistically significant.

The success of the Sherlock field test has resulted in high-level support for the BJS program from within Tactical Air Command which employs the maintenance personnel whose training Sherlock targets. In order to get the tutor into Air Force maintenance work places, Sher-

lock is currently undergoing a hardware and software conversion which will allow the system to be delivered on standard PC hardware that is available in maintenance work centers. This conversion is being carried out at the University of Pittsburgh Learning Research and Development Center by researchers responsible for the original development of Sherlock. In addition to the need for delivering Sherlock on standard Air Force hardware, decisions regarding Sherlock's conversion have been driven by three primary concerns: first, the instructional features that led to the tutor's success must be better understood and retained; secondly, the tutor's limitations must be explicated and reduced; and finally, the resultant tutor courseware must be maintainable by Air Force personnel.

INSTRUCTIONAL FEATURES OF THE AVIONICS TROUBLESHOOTING TUTOR: TROUBLESHOOTING PRACTICE IN A SIMULATED, SUPPORTED WORK ENVIORNMENT

The instructional features of Sherlock that appear to be responsible for the dramatic learning gains are associated with the simulated, supported practice environment that the tutor provides. Specifically, opportunities for realistic practice, feedback to foster the development of a mental model of an electronic test, menus that support the development of goal-oriented activity, and multiple levels of hints from Sherlock's coach are of particular interest.

One of Sherlock's most important instructional features is that it provides students with the opportunity to practice solving realistic troubleshooting problems in a simulated but supported work environment. Figure 2 shows the tutor display as it appears to the student upon presentation of a troubleshooting problem. The context of the problem is established by presentation to the student of a scenario that technicians might encounter on the job. The problem is thus presented in much the same way that a real problem would present itself in the shop. The work environment of the shop is also represented in the form of a simulated test station, a unit from the jet that is being tested (referred to as a line replaceable unit or LRU), and a test package connecting the LRU to the test station. The simulated dimensions of the equipment are primarily the external controls of test station drawers rather than their internal functional behavior. Front panels of test station drawers were graphically simulated to appear as similar to the real work environment as possible, and indicators and controls were functionally simulated to allow manipulation by the student for the purpose of performing tests and taking measurements. Within the test station, measurements are taken by selecting test points on schematic diagrams displayed on Sherlock's screen. Measurement values have been prespecified, however, and do not result from an underlying deep simulation of the device (i.e., test equipment and LRU).

In most of Sherlock's problems, as in the real shop environment, a corrective action or "fix" called out by the technical orders for a failed test step rarely fixes the problem. It is at this point then that students must begin to think on their own to develop a plan for isolating the fault. This requires relating the failed test to a mental model of the system as it was presumed to be functioning at the time of the fail. This envisioning process involves representing components of the system that were active during the test, and the flow of information through these components. Figure 3 illustrates an abstract model of an electronic test which can be used to characterize any circuit path that the student might have to investigate. A stimulus signal is generated by one of the drawers in the test station, and sent to a routing device which routes the signal through the test package and the LRU. The LRU responds to the input signal and produces an output which is sent back to the test station and routed to a measurement device (Lesgold, Lajoie, Bunzo, and Eggan, 1988). In relating this abstraction to a particular test, the student is encouraged to identify the active circuit path for that test. The model of the test thus provides a structure for the organization of the student's declarative knowledge of the system and constrains the search for the fault.

The tutor is also directed at the development of goal structures for investigating the equipment, procedural knowledge of specific troubleshooting actions, and additional strategic knowledge required to inform decision making during problem solving (Gott, 1989). Sherlock's action menu, shown at the right-hand side of the display reproduced in Figure 2, allows students to choose which area of the equipment they want to investigate, and to select the procedures for doing so. Some of these menu selections have additional choices embedded within them representing further decisions that the technician must make in pursuing a particular solution path. The menus serve to structure the problem-solving process and facilitate the apprentice's development of a conceptual model of the task. Thus, for example, in testing an LRU that has come in from the flight line, the student must access the technical order that describes the test procedures for that particular LRU, set up the drawers as

instructed for each test on the LRU (e.g., wiring integrity tests, power short tests, resistance tests, etc.), and run and interpret each test. If a test fails, the technical order might call out a suggested fix for the fault, and the student is encouraged to try that fix before investigating other components as the cause of the failed test. Other procedural choices represented in Sherlock's action menus include selection of test points, selection of components to be replaced, swapping suspected bad components, checking connections, etc. Sherlock thereby provides a simulated learning and practice environment so that technicians can exercise the skills they must use in the real work environment. Moreover, Sherlock embodies a coach or master technician to foster apprenticeship learning with feedback and general problem solving assistance.

Sherlock's coach offers external support in the form of hints that are provided when the student asks for help. The hints, like the action menu choices, are tied to the goal structure of fault isolation tasks, and vary according to type and level of explicitness. Hint type is related to the student's current troubleshooting activity and specifies, for example, where to take a measurement or how to interpret a measurement already taken. The explicitness of the hint is determined by the student who can access up to five levels of increasingly directive hints, from a simple recap of past plans and actions, to detailed information concerning how to perform the next suggested action. Unsolicited intervention from the coach can also occur under certain circumstances, for example, if the student fails to turn off a hazardous voltage prior to extending a circuit card, or investigates a piece of equipment that was not being used when the test failed. Sherlock's hints are thus adaptive in the sense that hints received are dependent on the individual student's activity at the time the hint is accessed, and the desired level of assistance as indicated by the specificity of the hint requested.

Sherlock's instructional limitations result primarily from the fact that the tutor's curriculum is to a large extent prespecified. The problems presented to students and their sequence is the same for all students, regardless of their individual strengths and weaknesses. Although the tutor evaluates students' problem solutions and highlights their strengths and weaknesses in post-problem feedback, this diagnostic capability is not exploited to provide problems that are particularly adapted to the individual student's current level of skill. This lack of adaptiveness exists because the tutor does not possess the capability of generating new problems on line in the course of tutoring. Further, on-line diagnosis of students' troubleshooting is not robust enough to determine the appropriate type and level of hint to provide when a student asks for help. While the presentation of hints is adaptive in the sense described earlier, the hints themselves have been prespecified and the principles that determine hint content and guide Sherlock's decisions to intervene are not as yet clearly established. In the next version of the tutor, simulation will be deeper in the sense that a set of circuits will be functionally simulated and the electronic tests performed on these circuits will be modelled. This simulation will provide the basis for improved student modelling and diagnosis, on-line problem generation, and more principled explanations and student feedback (see following section for a more complete discussion of how these improvements will be implemented).

THE AVIONICS TROUBLESHOOTING TUTOR II

The next generation of Sherlock is presently under development, with the concerns described above providing the foci for the effort. The goals include delivery of instruction on accessible, cost-effective hardware, simplification of tutor development and maintenance by Air Force personnel, and increased adaptiveness in instruction, including improved student diagnosis and on-line problem generation capability.

Sherlock was originally implemented in the Xerox Lisp environment in order to take advantage of its large internal memory capacity and superior graphics capabilities. The idea was to first test the validity of the cognitive models and theory underlying Sherlock's design utilizing optimal computer hardware. Now that the theoretical and empirical bases of Sherlock have been tested and supported, we must consider ways of delivering the tutor on a scaled-down system without sacrificing essential performance characteristics of more powerful machines. The basic configuration of the Avionics Troubleshooting Tutor II is depicted in Figure 4. The system consists of an 80386-based PC with one MB of internal memory and two to three MB of expansion memory. The PC is connected, via an RS-232 cable, to a video disc player which stores video images to be displayed on a 20-inch multiscan, high-resolution monitor. The PC is equipped with VGA graphics and a superimposer board for overlaying computer graphics on video images.

The use of video in displaying the work environment provides several advantages over computer graphics. First, by using

video images of the real test equipment. Sherlock's feature of providing a realistic work environment is retained, and in fact, enhanced with concrete visual representations of actual physical equipment. Figure 5 shows the front panel of one of the test station drawers in the original Sherlock. The time required to develop such detailed graphics and all possible configurations of each front panel, not to mention their storage requirements, represents a significant investment of resources to achieve work environment realism in Sherlock I. That investment will be significantly reduced via the use of video in Sherlock II. In that version of the tutor, computer graphics will be used almost exclusively in the menus, with a resultant savings in development time and storage. Second, because the Air Force maintenance community currently uses interactive video in developing (and delivering) its maintenance training, instructional designers who will ultimately maintain the tutor are already familiar with the technology.

The converted tutor is being developed in the Smalltalk V286 software environment which will allow significant savings in development time and facilitate the implementation of more adaptive instruction in the form of on-line hint and problem generation. Recall that the converted tutor will employ simulated circuits. The object-oriented environment provided by Smalltalk was chosen since it implements sophisticated class structures with asynchronous messaging between objects, thereby allowing for the simulation of complex systems. The environment also reduces development time because the object class structure defined by the programmer determines the properties of objects within a class and the operations that can be applied to them. Put another way, objects inherit the properties and operations (or methods) of their class which are defined only once for the entire class rather than for each object within the class. The reduction in development time thus results from the ability to, in effect, use a given piece of code for multiple purposes.

The implementation of a class structure is illustrated by the simulated circuits being developed for Sherlock II, and the electronic tests which operate on the circuit. Each instance of a test involves four elements: a signal source, an LRU (the unit being tested), a measurement device, and a circuit path. Although different tests may use different circuit paths, stimulus sources, etc., all instances of each element have certain behaviors in common. For instance, a broken wire in any circuit path will, in general, cause an ohms measurement to indicate infinite resistance. If, in a particular circuit, the wire was shorted to another wire, the reading might be different. It is only under these unusual types of conditions that additional code must be written to override the behavior that defines circuits in general. Thus, rather than coding each circuit to be used in the tests independently, all circuits will share code that defines their common behaviors. The ability to capture the general properties of objects will provide the basis for rule-based problem generation and hint generation in Sherlock II. Given that problems and hints can be created by rule, then they can be generated during the course of the tutoring session in a way that is responsive to the individual's troubleshooting strengths and weaknesses, thus providing more adaptive instruction.

The object-oriented programming environment also promotes maintainability of the tutor because it lends itself to modularization. Modular code makes the structure of the program clearer, thus facilitating modifications by programmers who were not involved in the tutor's original development. Modules that do not require modification for new versions of the tutor or for other tutors being developed for different maintenance job specialties can also be easily transported. The BJS Program is currently conducting a training needs assessment to determine the type of programming expertise required for maintenance of intelligent tutors developed in the object-oriented languages so that this task can be performed by Air Force personnel.

CONCLUSIONS

In addressing the needs of the Air Force maintenance community, the Basic Job Skills Program has benefitted from methodological and theoretical advances in cognitive science. These benefits are reflected in, for instance, the cognitive task analysis procedure which was used to inform Sherlock's design (Gott, 1987) and the increasingly comprehensive models of troubleshooting performance that the results of these analyses yield. To the extent that the cognitive approach to Sherlock's design contributed to the tutor's effectiveness, then an important future goal for the BJS Program will be to make this technology available to nonscientists in the Air Force who are responsible for instructional design and maintenance of educational courseware. Steps toward this goal include the development of maintainable software that is compatible with standard hardware, and the attempt to gain a better understanding of the instructional approach required in teaching a complex skill like troubleshooting.

REFERENCES

Gitomer, D.H. A COGNITIVE TASK ANALYSIS OF A COMPLEX TROUBLESHOOTING TASK. Unpublished doctoral dissertation, University of Pittsburgh, Pittsburgh, PA, 1984.

Gitomer, D.H. "Individual differences in technical troubleshooting." Human Performance, Vol. 1, No. 2, 1988, 111-131.

Glaser, R., Lesgold, A., Lajoie, S., Eastman, R., Greenberg, L., Logan, D., Magone, M., Weiner, A., Wolf, R., and Yengo, L. "Cognitive task analysis to enhance technical skills training and assessment," (Contract No. F41689-83-C-0029). Brooks AFB, TX: AFHRL, 1985.

Gott, S.P. "Tutoring electronic troubleshooting in a simulated maintenance work environment," Paper presented at the First Annual Workshop on Space Operations Automation and Robotics (SOAR), NASA/ Johnson Space Center, Houston, TX, August 1987.

Gott, S.P., Bennett, W., and Gillett, A. "Models of technical competence for intelligent tutoring systems," JOURNAL OF COMPUTER-BASED INSTRUCTION, Vol. 13, No. 2, 1986, 43-46.

Gott, S.P. "Apprenticeship instruction for real-world tasks: The coordination of procedures, mental models, and strategies," REVIEW OF RESEARCH IN EDUCATION, E.Z. Rothkopf (Ed.), Vol. 15, 1989, 97-169.

Lesgold, A., Lajoie, S.P., Bunzo, M., and Eggan, G.M. "Sherlock: A coached practice environment for an electronics troubleshooting job," Paper presented at the J.S. Macdonald Foundation Conference on CAI and ITSs, Carnegie-Mellon University, Pittsburgh, PA, 1988.

Nichols, P.D., Pokorny, R., Jones, G., Gott, S.P., and Alley, W.E. "Evaluation of an avionics troubleshooting tutoring system," Air Force Technical Report, Air Force Human Resources Laboratory, Brooks AFB, TX, in preparation.

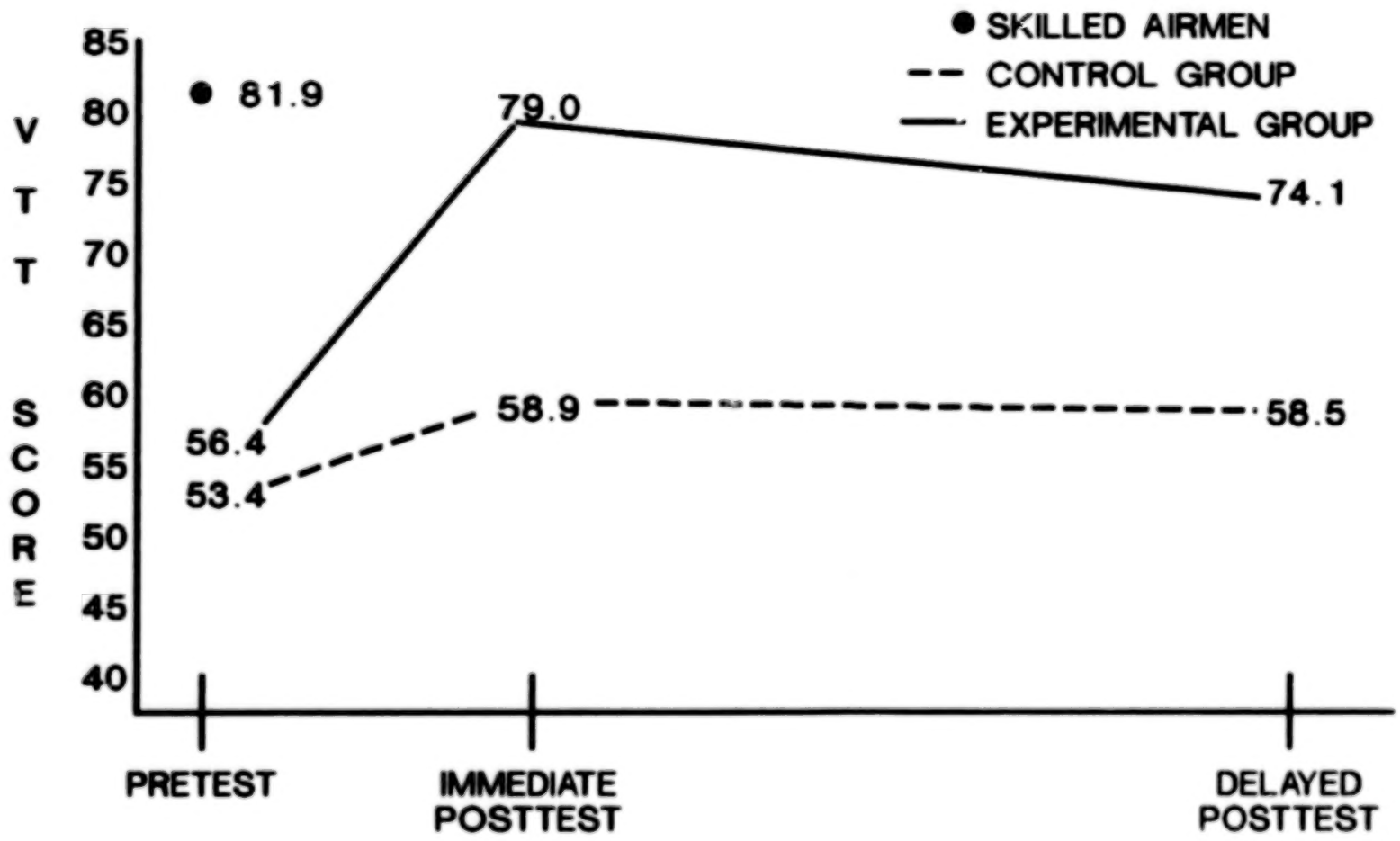


FIG. 1. RESULTS OF FIELD TEST

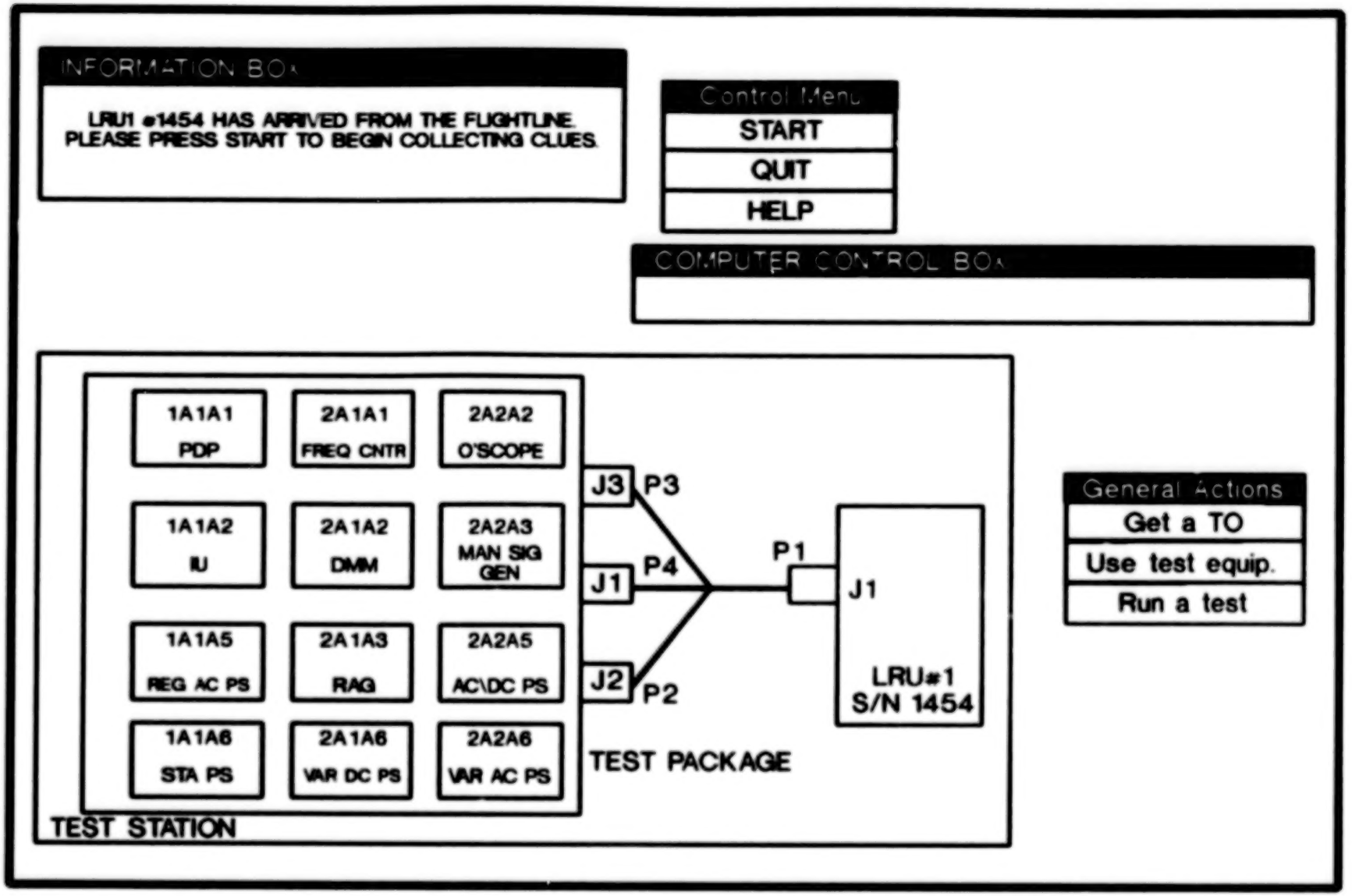


FIG. 2. WORK ENVIRONMENT DISPLAY

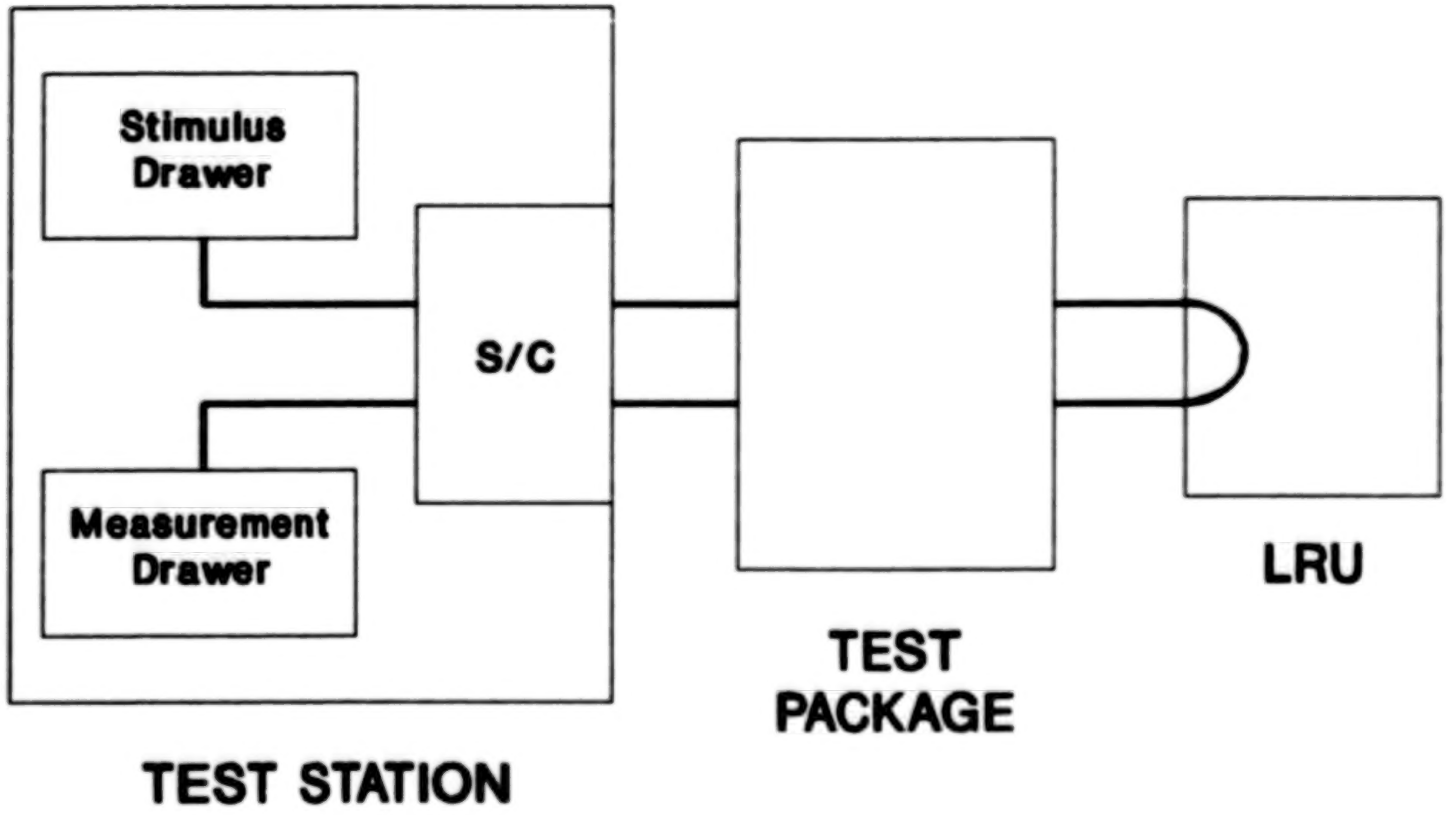
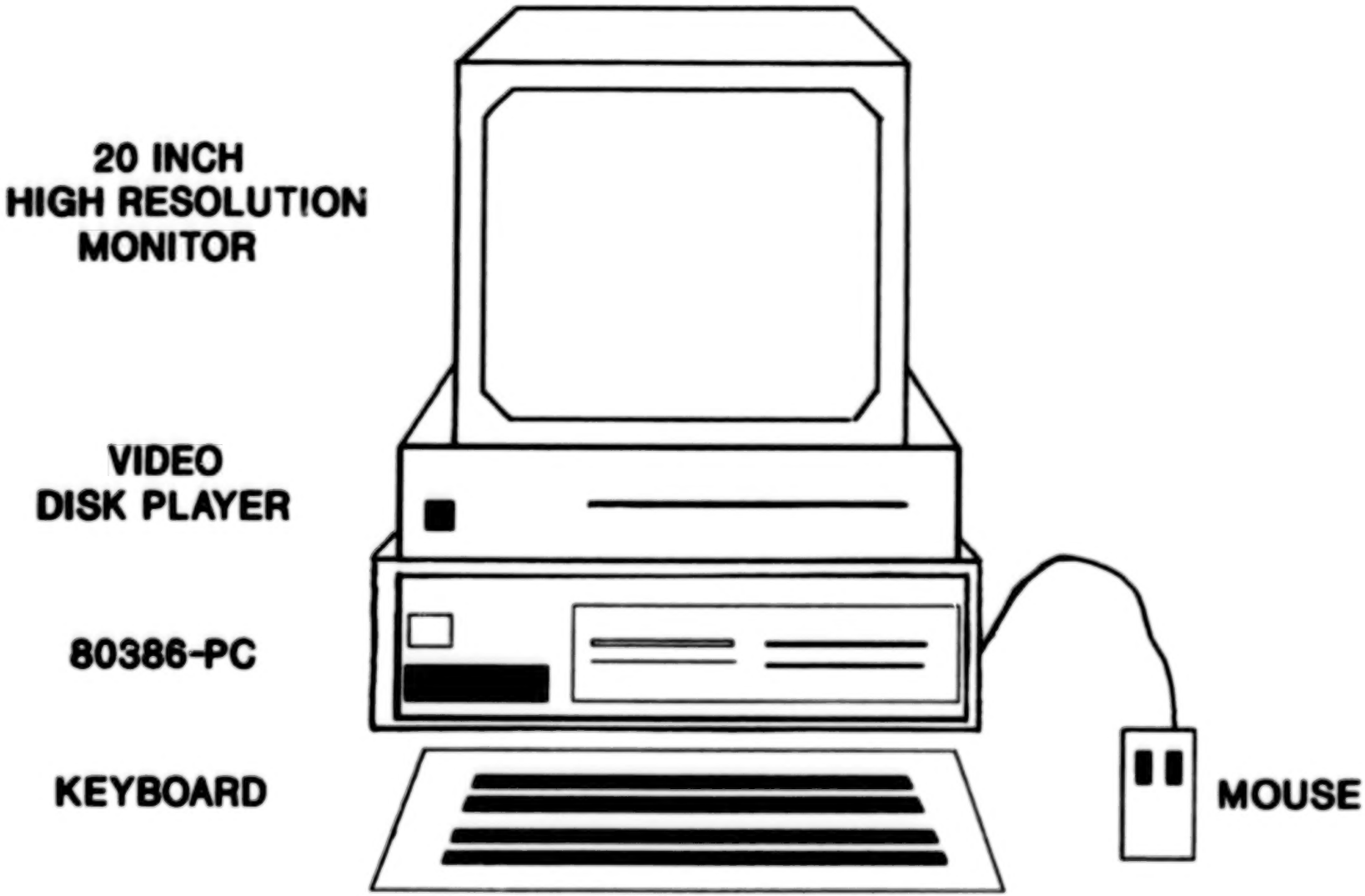
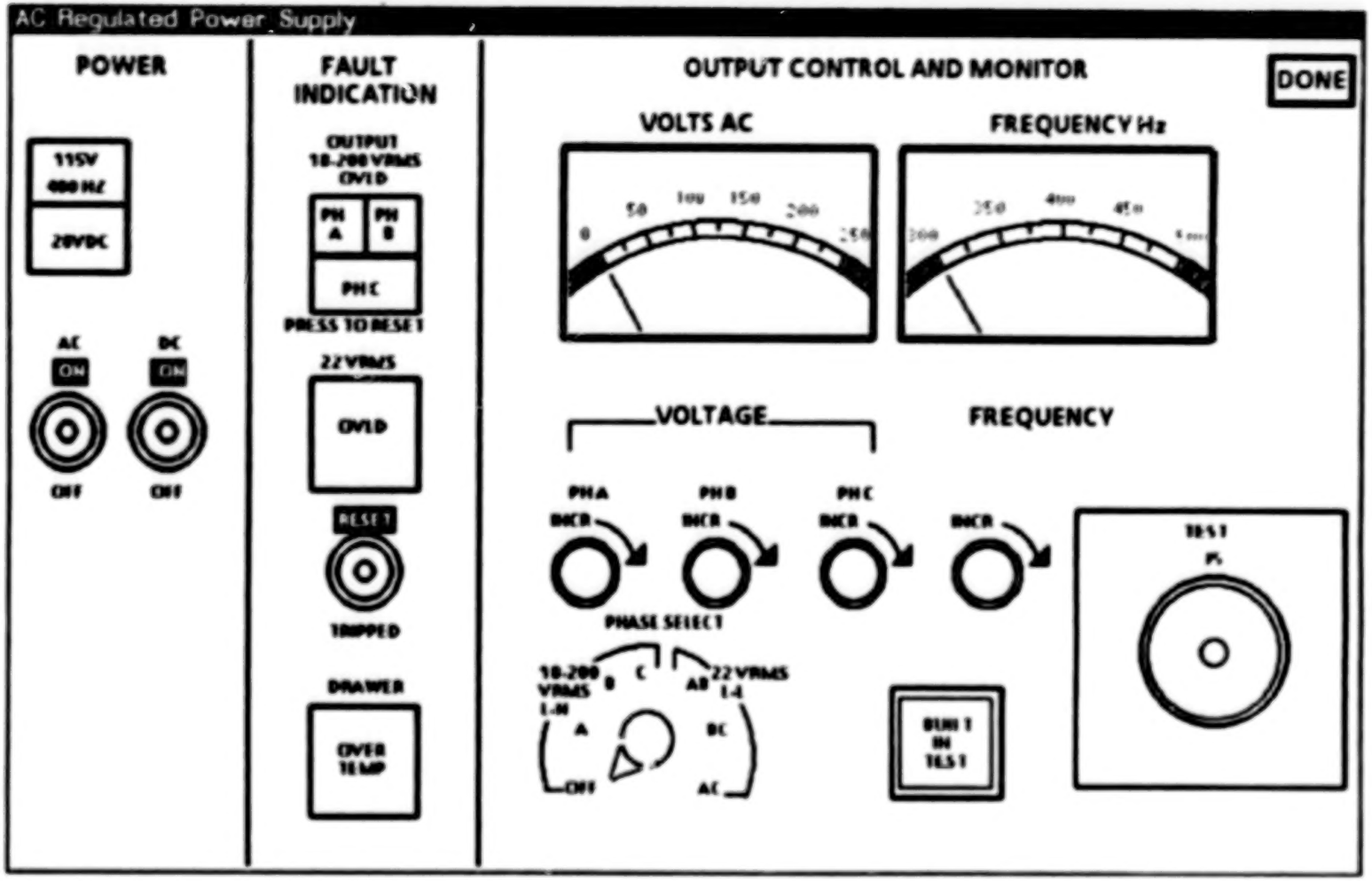


FIG. 3. AVIONICS EQUIPMENT CONFIGURATION (SIGNAL PATH)



602

FIG. 4. CONFIGURATION OF
AVIONICS TROUBLESHOOTING TUTOR II



603

FIG. 5. FRONT PANEL DISPLAY OF A TEST STATION DRAWER IN SHERLOCK

A MODEL FOR A KNOWLEDGE-BASED SYSTEM's LIFE CYCLE

Peter A. Kiss
BDM International, Inc.
950 Explorer Boulevard
Huntsville, AL 35806

ABSTRACT

The American Institute of Aeronautics and Astronautics has initiated a Committee on Standards for Artificial Intelligence. Presented here are the initial efforts of one of the working groups of that committee. The purpose of this paper is to present a candidate model for the development life cycle of Knowledge Based Systems. The intent is for the model to be used by the Aerospace Community and eventually be evolved into a standard.

The model is rooted in the evolutionary model, borrows from the spiral model, and is embedded in the standard Waterfall model for software development. Its intent is to satisfy the development of both stand-alone and embedded KBSs. The phases of the life cycle are detailed as are and the review points that constitute the key milestones throughout the development process. The applicability and strengths of the model are discussed along with areas needing further development and refinement by the aerospace community.

1.0 INTRODUCTION

This paper presents a model for the life-cycle development of knowledge-based systems. The Artificial Intelligence Software Engineering (AISE) Model is an outgrowth of an effort by an AIAA committee on standards for AI. This committee was convened in early 1989 to explore the potential for developing various standards or guidelines for AI. Three working groups were formed to explore definitions and lexicon compilation, tools standardization feasibility, and development of life-cycle guidelines. The course of our approach is to develop candidate guidelines, disseminate to the community for feedback, and slowly evolve to standards as acceptance of the products grows. It is in that spirit that this paper presents the AISE model to the aerospace community for its feedback.

During the past ten years, the Knowledge-Based System (KBS) branch of Artificial Intelligence (AI) has matured considerably. Many small prototype systems have been successfully developed and implemented. Larger KBSs are much more complex and have been implemented at a slower rate. The organizations at the leading edge of using AI, ones that have been developing KBSs and applying them, are looking at the integration of KBSs into the mainstream of their computing environments. This is taking a more traditional total systems approach to AI, making the KBS an integral part, not a standalone tool. With the perspective of a systems approach comes the need for more rigorous development and integration methodologies. This need, coupled with general community's desire to control costs and schedules, is the impetus for the AISE model.

The objective of the AISE model is to provide a flexible framework for the development of a KBS (either standalone or integrated) with meaningful milestones and reviews that support the control of technical, cost, and schedule dimensions of a program. To achieve this objective, the model borrows the best attributes of the evolutionary software development model and some of the spiral model concepts and embeds them in the Waterfall model for software development.

2.0 SOFTWARE DEVELOPMENT MODELS

Several basic phases are inherent parts of any software (including AI) development program: Problem conceptualization/definition; system design; system development; testing; integration; and maintenance and enhancement. The

sequence in which these are carried out, the amount of emphasis/effort given each phase, and the controls associated with execution of the work combine to define a life-cycle model.

The Waterfall model, shown in Figure 1, is the most widely used in one variation or another. In the concept definition phase, studies and trades are conducted to define the system to be built. As a result, this phase culminates in a minimum of system requirements, top-level design specifications, and an operational concept. Next, a Preliminary Design Phase fleshes out the specifications and top-level design. Interfaces and data bases are specified, critical methods (such as special algorithms) are addressed, and test plans are conceived. The Preliminary Design is followed by a Detailed Design phase that finalizes the design and specifications. Simulations and prototyping are used to test the design, and test plans and operations

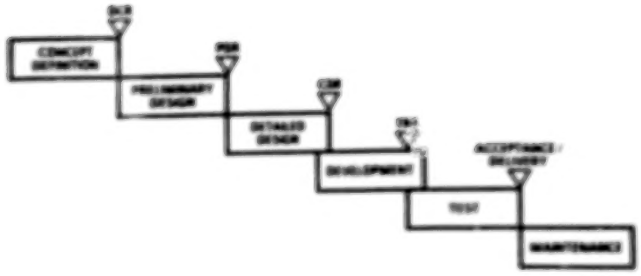


Figure 1. Standard Software Development Life-cycle

manuals are developed. Once design is complete, the software is coded/developed according to it and the specifications. As the software components are developed, they are tested and hierarchically validated and integrated to form the overall system. Once the system is accepted by its users, there is usually a long life of maintenance and upgrades during its operation.

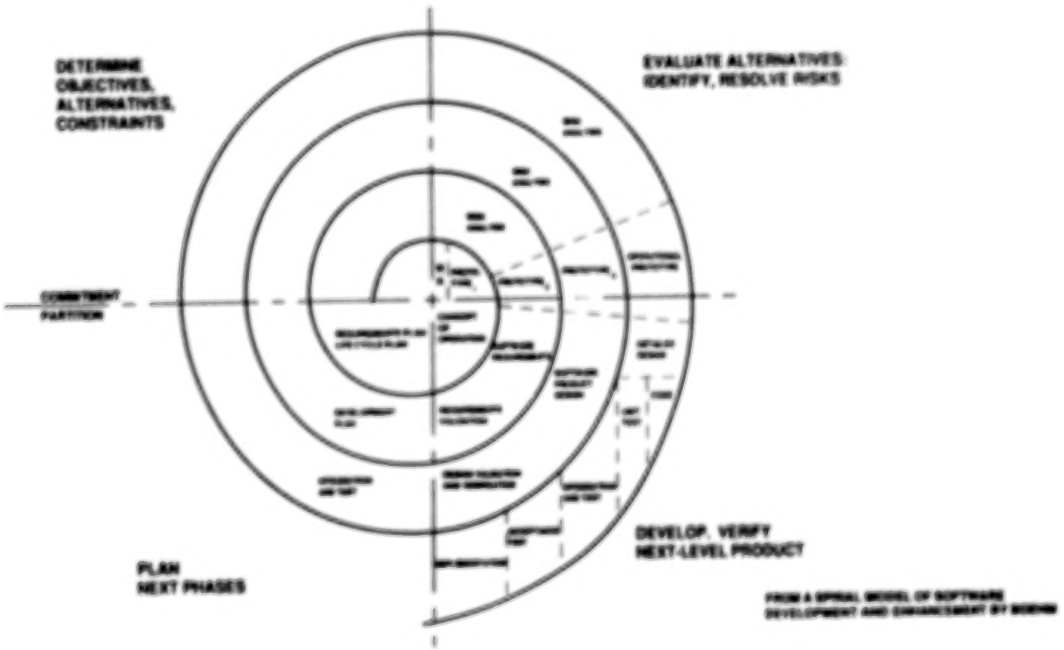


Figure 2. Spiral Model of the Software Process

The Spiral model, developed at TRW and shown in Figure 2, follows a different sequence. Once a problem is conceived, a series of prototypes is used to address the areas of highest risk in order of difficulty. Once all the parts of the system are well understood and the prototypes have developed a preliminary design, this model picks up the back phases of the Waterfall model to finish the product. A key characteristic of the Spiral model is the non-uniform maturation of system parts.

Another methodology for software development, one often used for AI, is the Evolutionary Model. Under this model, software is developed and tested incrementally for most of its life cycle.

Figure 3 provides a comparison of the models discussed. Given are the most appropriate situations for the application of each of the models, along with their strengths and weaknesses. If we examine the chart in light of some key characteristics of an aerospace KBS

development, we are lead to the conclusion that a hybrid model is needed. Three characteristics of KBS development are essential for aerospace applications: 1) There is usually uncertainty in the scope of the problem and its appropriate solution; 2) the knowledge engineering process is inherently an evolutionary process; and 3) projects tend to have tight cost and schedule budgets. These three items point to a model that has flexibility and is evolutionary in nature while at the same time has a firm structure to control the development process. The Artificial Intelligence Software (AISE) model is designed to meet these needs.

MODELS	APPLICABILITY	STRENGTH	WEAKNESS
WATERFALL (SPECIFICATION DRIVEN)	• LARGE SCALE DEVELOPMENT • WELL DEFINED PROBLEMS • CONSTRAINED RESOURCES • GOVERNMENT REQUIREMENT	• RIGOROUS STRUCTURE • WORKS TO CONSTRAINTS • GOOD DEVELOPMENT VISIBILITY	• DIFFICULT TO CHANGE • UNIFORM PROGRES OF ALL COMPONENTS • DOES NOT ACCOMODATE EVOLUTIONARY DEVELOPMENT
SPIRAL (RISK DRIVEN)	• MEDIUM SIZE DEVELOPMENT • KNOWN RISKY AREAS • UNCONSTRAINED RESOURCES	• ACCOMODATES NON UNIFORM DEVELOPMENT • CONCENTRATE ON CRITICAL COMPONENTS • ADAPTIVE TO OTHER MODELS	• LIMITED COST AND SCHEDULE CONTROLS • LIMITED DEVELOPMENT OF MILESTONES AND REVIEWS • LIMITED SPECIFICATION AND DOCUMENTATION DEVELOPMENT
EVOLUTIONARY (PROTOTYPE DRIVEN)	• SMALL TO MEDIUM SIZE • ILL DEFINED PROBLEMS • UNCONSTRAINED RESOURCES	• INCREMENTAL BUILD • EASY TO CHANGE DIRECTION	• LIMITED COST AND SCHEDULE CONTROLS • LIMITED CONTROL OF REQUIREMENTS • DIFFICULTY SCALING UP • NO VISIBILITY INTO PROCESS

Figure 3. Software Development Models

3.0 ARTIFICIAL INTELLIGENCE SOFTWARE ENGINEERING (AISE) MODEL

The AISE model, shown in Figure 4, focuses on the KBS element of a system as an area of high risk. It drives the development to be at the same level of maturity for its components at each major milestone, thus providing for process control.

The AISE phases and their relations to each other are shown in Figure 4. In the following sections, we discuss the objectives, activities, and results of each phase.

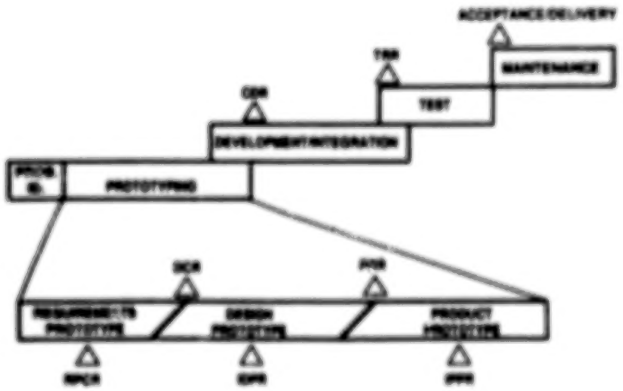


Figure 4. The AI Software Engineering (AISE) model

3.1 Problem Identification

Objectives: Analyze and define problem elements that are suitable for KBS solution.

Activities:

1. Isolate problem areas that are potentially suitable for KBS solution
2. Perform trades to determine whether KBS is the best solution compared to other techniques
3. Perform cost/benefit analysis
4. Draft development plans, including key participants needed

Results/Products: A well defined and justified KBS application with a plan for its development

3.2 Prototyping

Objective: Develop a full-capability prototype of the KBS element along with a detailed design for its target implementation

Activities: A series of three prototyping iterations and six reviews

1. Evolve a prototype to a full knowledge set
2. Test prototype during development
3. Develop documentation
4. Design target environment
5. Review prototyping progress

Results/Products: A fully developed product prototype of the KBS, the design for its target environment, and the associated support documentation.

The prototyping phase is the most critical in building a KBS, and accordingly it is the heart of the AISE model. The content and control of the work done in this phase will determine the success of the system being built. Figures 5, 6, and 7 show details of the review milestones associated with each of

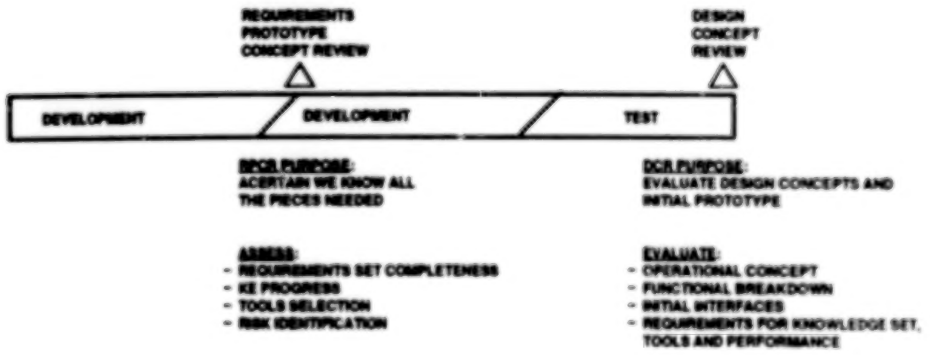


Figure 5. Requirements Prototype

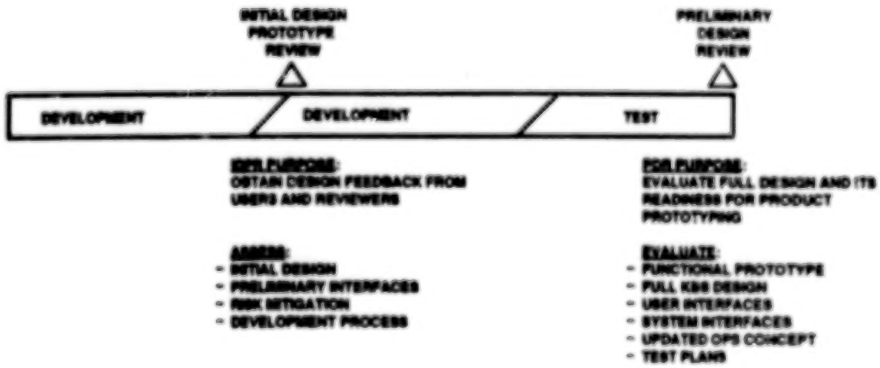


Figure 6. Design Prototype

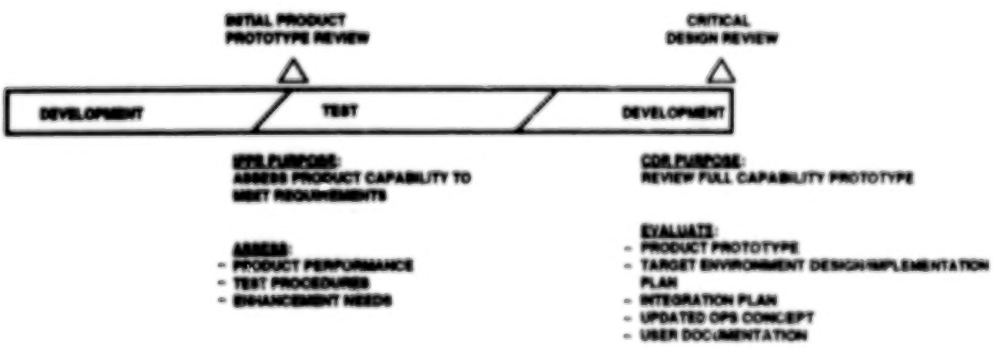


Figure 7. Product Prototype

the prototyping stages and the contents expected at each review.

3.3 Development/Integration:

Objective: Embed the KBS into its target environment.

Activities:

1. Port KBS to intended host environment (and, if applicable, language)
2. Integrate with system components that are external to the KBS through interfaces (I/F)
3. Implement integrated user I/F
4. Develop documentation

Results/Products: An integrated KBS in its target environment

3.4 Test and Evaluation

Objective: To ensure that the overall system works according to specifications and meets its requirements

Activities:

1. Perform hierarchical tests with greater levels of integration
2. Perform regression tests to check against standalone KBS prototype results
3. Evaluate overall system performance
4. Validate that the system meets requirements
5. Perform acceptance testing

Results/Product: Completed system ready for delivery to user

3.5 Operations and Maintenance:

Objective: Apply system to its intended use

Activities:

1. Routine operation of system
2. Debugging as required
3. Enhancements as the needs come up

Result/Product: A gracefully maturing system

The life-cycle of the AISE model has been planned to be compatible with the Waterfall model. This was done deliberately since many aerospace programs are mandated to use a variant of the Waterfall model (many are requested to use the 2167A standard). Figure 8 shows how the AISE model folds into the Waterfall. The review milestones align precisely with the completion of the prototyping phases and the two merge during the development phase.

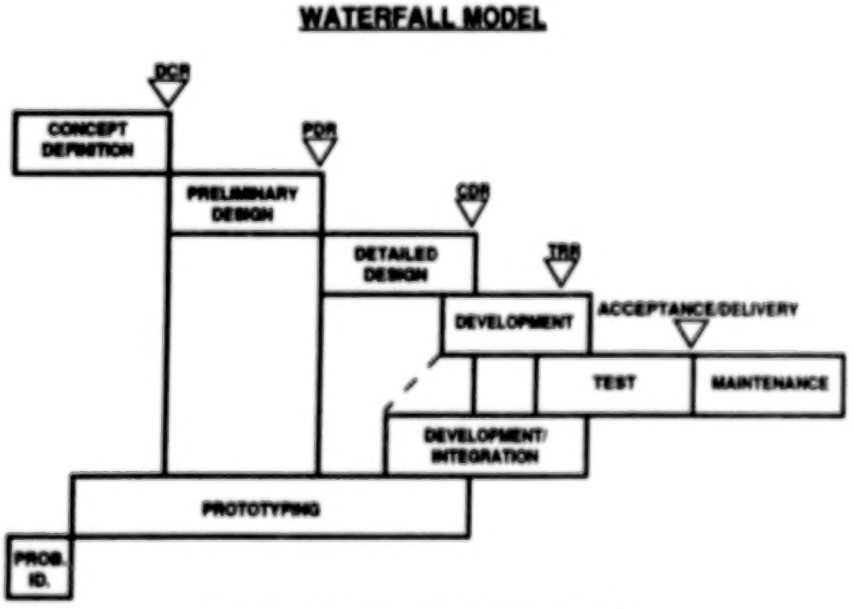


Figure 8. Integrated/Embedded Methodology

4.0 CONCLUSIONS

In order to integrate KBSs into the mainstream of software development and aerospace applications, a more rigorous development methodology is needed. The most popular software development models have been examined for their applicability and characteristics. A new hybrid, the AISE model, is proposed for KBS development. The AISE model provides flexibility up front for evolution of a knowledge base. At the same time, it provides visibility into development through a series of reviews. One of the features of the AISE model is the uniform (at milestones) development of all components of the KBS. This uniformity allows for meaningful development of requirements and specifications for the whole system, which in turn provides the mechanisms for technical, cost, and schedule controls. Finally, the AISE model can be neatly merged with the Waterfall model making the AISE model applicable and compliant with most Government software acquisition requirements.

5.0 ACKNOWLEDGEMENTS

This work is an outgrowth of the AIAA committee on standards for AI. The continued support of the following members is greatly appreciated: Robert Savely, Chuck Hall, Greg Swietek, Carol Russo, Guy Olney, Brad Allen, Mike Freeman, and Bruce Bullock. Also appreciated is the work of Nancy Seawell, with whom we formulated the early concepts of KBS development.

6.0 REFERENCES

1. B. Boehm, A Spiral Model for Software Development and Enhancement, ACM Sigsoft Software Engineering Notes, August 1988.

2. DOD STD 2167A and associated Software Specifications documents.

3. Dr. M. Freeman and P. Kiss, Issues in Management of Artificial Intelligence Based Projects, Fourth Conference on AI for Space Applications, November 1988.

4. F. P. Brooks, The Mythical Man-Month, Addison-Wesley, 1975

5. R. Pressman, Software Engineering, McGraw-Hill, 1987.

6. B. Boehm, Software Engineering Economics, Prentice-Hall, 1981.

KNOWLEDGE-BASED SYSTEMS(KBS) DEVELOPMENT STANDARDS A MAINTENANCE PERSPECTIVE

Capt John Brill, USAF
USAF Office of Logistics Technology Assessment
Wright Patterson AFB OH

611

OVERVIEW

- PURPOSE
- KNOWLEDGE-BASED SYSTEMS PERSPECTIVE
- CONVENTIONAL COMPUTING/SOFTWARE EXPERIENCE
- KBS STANDARDIZATION
- SUMMARY

PURPOSE

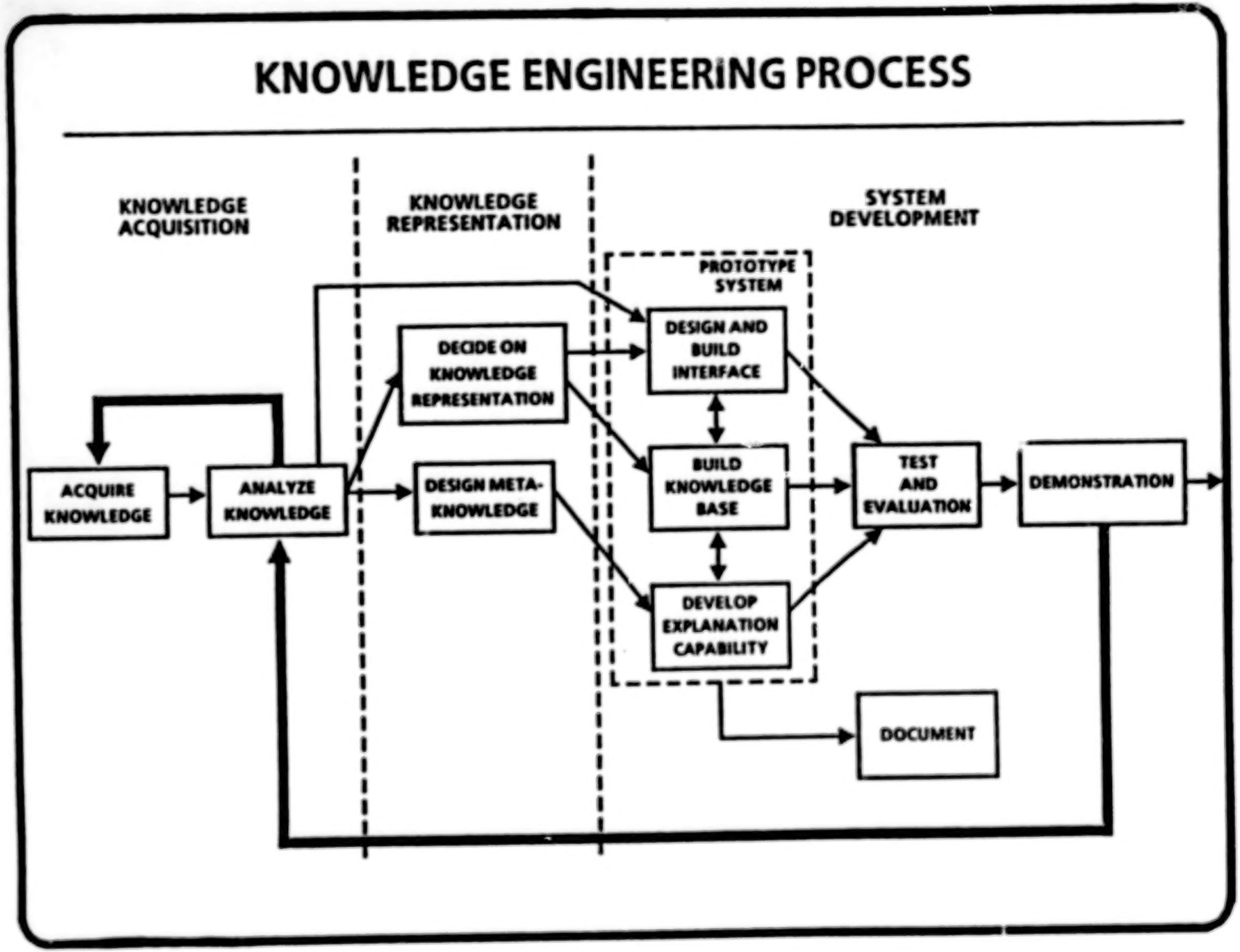
- IDENTIFY KBS STANDARDIZATION NEEDS/ISSUES RELATED TO SUPPORT OF KNOWLEDGE-BASED SYSTEMS

KNOWLEDGE-BASED SYSTEMS PERSPECTIVE

KBS GENERALIZES APPLICATION OF COMPUTING/SOFTWARE IN SYSTEMS

- **Broader class of problems is addressed vs traditional efforts**
- **Non-algorithmic solutions are made practical**
- **Higher-order language/representation provide better user interface**

...Capabilities which must be supported


KNOWLEDGE ENGINEERING PROCESS
KNOWLEDGE ACQUISITION
KNOWLEDGE REPRESENTATION
SYSTEM DEVELOPMENT
PROTOTYPE SYSTEM
ACQUIRE KNOWLEDGE
ANALYZE KNOWLEDGE
DECIDE ON KNOWLEDGE REPRESENTATION
DESIGN META-KNOWLEDGE
DESIGN AND BUILD INTERFACE
BUILD KNOWLEDGE BASE
DEVELOP EXPLANATION CAPABILITY
TEST AND EVALUATION
DEMONSTRATION
DOCUMENT

KNOWLEDGE-BASED SYSTEMS PERSPECTIVE (CONT'D)

KBS CAPABILITIES OF GREATEST INTEREST TO AFLC

- Improved consistency in decision making
- Force status, readiness, and system recovery
- Support/maintenance automation for major systems
- Inspection, diagnosis, and repair assistance
- Inventory, supply and distribution tracking
- Automated document generation/update/delivery
- Management system database streamlining

...KBS is central to improved support systems

KNOWLEDGE-BASED SYSTEMS PERSPECTIVE (CONT'D)

KBS PRESERVES FLEXIBILITY TO REFINE APPLICATIONS

- **Emphasizes ongoing refinement of mission processes and criteria**
- **Drives clean engineering practice, e.g. separation of control, processes, knowledge and special purpose modules**

...KBS expand the scope of support to include continued refinement of mission tasks and knowledge bases

617

KNOWLEDGE-BASED SYSTEMS PERSPECTIVE (CONT'D)

MAJOR SYSTEM SUPPORT/KBS APPLICATIONS

- Supportability
- Support System Design

IMPROVEMENT TO FIELDED SYSTEMS

- Post Deployment Modification
- Life Cycle Support Environments

ORGANIZATIONAL INFRASTRUCUTRE

- Data Systems
- Task Automation

...Standards approach should recognize similarities, differences

KBS SUPPORT - WHAT'S DIFFERENT?

- CONTINUED UPDATE/REFINEMENT OF KNOWLEDGE BASES
- SUPPORT OF NEW HIGH-LEVEL DESCRIPTION LANGUAGES
- NON-TRADITIONAL SYSTEM CONFIGURATIONS
- MORE COMPLICATED POTENTIAL FAILURE PROPERTIES

CONVENTIONAL COMPUTING/SOFTWARE EXPERIENCE

- POLICY - AFR 800-xx, AFR 700-xx
- WORKING GROUPS - CRWG, IMWG
- LIFE CYCLE PLANNING
- PHASED ACQUISITION CRITERIA

CONVENTIONAL COMPUTING/SOFTWARE EXPERIENCE

TECHNICAL STANDARDS/GUIDES

- Process
- Architecture
- Baseline Management/CM
- Quality Eval
- Languages
- Risk Assessment
- Environments
- Requirements Screeening

...Software support treated as routine CM and modification separate from hardware and system engineering

CONVENTIONAL COMPUTING/SOFTWARE EXPERIENCE (CONT'D)

AREAS WEAKLY ADDRESSED IN CONVENTIONAL APPROACH

- **Support Process and Support Analysis**
- **System Engineering and Integration**
- **Network/Integrated-System Baselines and Command/Control Engineering**

...KBS will encounter the same problems unless we do something about it

SUPPORTABILITY PROGRAM REQUIREMENTS

- OPERATIONAL REQUIREMENTS, CRITERIA MUST INCLUDE (AF)
- COORDINATE SUPPORT CONCEPT BEFORE MSII (AF)
- COMPUTER RESOURCES SUPPORT AN ELEMENT OF ILS (DOD)
- INTEGRATE AF ILS FOR SDI (SDIO, AF)
- MCCR LIFE CYCLE PLAN, SIGNED BEFORE MS II (DOD)
- SUPPORTABILITY A CO-EQUAL ACQUISITION PRIORITY (DOD)
- USE CENTRALIZED INTEGRATION SUPPORT (MAJCOMS)
- ACQUIRE AND DELIVER SUPPORT CAPABILITY FOR BLUE-SUIT (AF)
- PROVIDE SUPPORTABILITY ASSESSMENT/PROGRAM REVIEWS (AF)
- SUPPORTABILITY R&D INCLUDING SOFTWARE (SDIO)

(Network role not uniformly addressed in P&G)

623

SUPPORTABILITY PROGRAM TASKS
AFR 800-8, 800-14, AFLC/AFSCP 800-34

- SUPPORTABILITY PROGRAM MANAGEMENT
  - PLANNING (CRLCMP) AND COORDINATION (CRWG)
  - PROGRAM/TECHNICAL REVIEWS & STATUS/RISK ASSESSMENT
- DEFINE SUPPORTABILITY OBJECTIVES, CRITERIA, & CONSTRAINTS
  - ARRIVE AT SOUND ENGINEERING DEFINITION OF BASELINES AND TESTABLE SUPPORTABILITY CRITERIA
- SUPPORTABILITY COVERAGE IN TECHNICAL REQUIREMENTS & TRADES
  - ENGINEERING INTEGRITY OF MISSION/SYSTEM BASELINE
- SUPPORTABILITY IN ACQUISITION REQUIREMENTS
  - ACQUIRE AND DELIVER SUPPORT CAPABILITY FOR BLUE SUIT OPS. AND POST DEPLOYMENT SUPPORT, FOR ALL SUPPORT PROCESSES
- LIFE CYCLE SUPPORT RESOURCES/LCC
  - IDENTIFY POST-ACQUISTION AND TOTAL RESOURCE REQMTS AND DRIVERS, E.G. SUPPORT CYCLES, TASKS, SKILLS, LOADS
- SUPPORTABILITY EVALUATION AND FEEDBACK
  - THRU EACH ACQUISITION/LIFE CYCLE PHASE

... BASELINE DEFINITION & EVALUATION IS CENTRAL

624

CHRONIC PROBLEM AREAS

- Support of burgeoning software/hardware inventory
- Poor requirements definition and traceability
- Underscoped costing, sizing, and risk assessment
- Integration problems and underscoping of size and effort in large systems/networks
- Incompatibilities of multiple languages, methods, conventions, and environments

...KBS, as a further expansion of 'software' role in systems, may be a lot tougher support problem than past software

KBS STANDARDIZATION

REASONS FOR KBS STANDARDS/GUIDELINES

- Simplify post-deployment support needs/structure
- Prevent unneeded proliferation of KBS products
- Ensure technical integrity of KBS applications/interfaces
- Prevent unproductive differences in baseline management, methods, and support strategies

626

KBS STANDARDIZATION

CRITICAL KBS STANDARDIZATION NEEDS

- KBS requirements definition and management in systems
- KBS costing, sizing, and risk assessment in systems
- KBS coverage in procurement definition/review
- KBS evaluation/verification/certification including criteria for supportable KBS
- KBS life cycle integration in systems (with conventional computing/software)

KBS STANDARDIZATION

Standards which span all life cycle phases

- **High-level conventions/languages for KBS and systems**
- **KBS support process in systems**
- **KBS supportability tasks, criteria including LSA/support analysis**
- **Selection criteria for KBS tools, environments**

KBS STANDARDIZATION ISSUES

- KBS PROLIFERATION IN ABSENCE OF STANDARDS
- EXISTING COMPUTING/SOFTWARE STANDARDS ADEQUACY
- SYSTEM INTEGRATION OF KBS
  - Impacts to performance, fault response, support requirements
  - Transition from legacy software or non-standard KBS
- SUPPORT PROCESS/KBS IN LOGISTICS STANDARDS
- KBS/OTHER ENGINEERING DISCIPLINES INTERFACE AND INITIATIVES - CIM/CASE, CALS, LSA,
- PARALLEL PROCESSING, ADA, etc.

SUMMARY

- KBS involves broader scope of concern than traditional computing/software
- Chronic problems for conventional computing promise to be even more difficult for KBS
- Standardization is essential to practically use KBS for defense systems
- Support-related issues and needs have been identified

630

ARCHITECTURE DESIGN AND IMPLEMENTATION STANDARDS FOR MEDIUM-TO-LARGE SCALE KNOWLEDGE-BASED APPLICATIONS

M. Stock
AI Technology, Inc.

(Paper not provided by publication date.)

ACES - Space Shuttle Flight Software Analysis Expert System

R. Scott Satterwhite
IBM - Systems Integration Division
3700 Bay Area Blvd. MC 6206A
Houston, TX 77058
(713) 282-7765

BACKGROUND

ACES (Analysis Criteria Evaluation System) is a knowledge based expert system that automates the final certification of the Space Shuttle onboard flight software. Guidance, Navigation and Control of the Space Shuttle through all its flight phases are accomplished by a complex onboard flight software system. This software is reconfigured for each flight to allow thousands of mission-specific parameters to be introduced and must therefore be thoroughly certified prior to each flight. This certification is performed in ground simulations by executing the software in the flight computers. Flight trajectories from liftoff to landing, including abort scenarios, are simulated and the results are stored for analysis. The current methodology of performing this analysis is repetitive and requires many man-hours. The ultimate goals of ACES are to capture the knowledge of the current experts and improve the quality and reduce the manpower required to cerfity the Space Shuttle onboard flight software.

FLIGHT SOFTWARE TEST DATA ANALYSIS

The volume of data required to certify the flight software for each mission is enormous. Between 15 and 25 simulations are run for each mission. Each simulation produces over 2 million plotted data points and over 30,000 lines of printed output. Analysis consists of manual evaluation of:

1. Sequences of flight activities
2. Cockpit CRT displays
3. Plotted flight software and simulator parameters

ACES automates the analysis of this data by using the same methodology that the experts use.

KNOWLEDGE ENGINEERING AND IMPLEMENTATION

The first phase of knowledge acquisition began by documenting the criteria that the experts use to analyze each of the testcases. This knowledge was written as pass/fail criteria for the sequences, displays and parameters in each test case. They were written such that they could be used manually by inexperienced analysts. Concurrently, automation techniques were evaluated and prototyped.

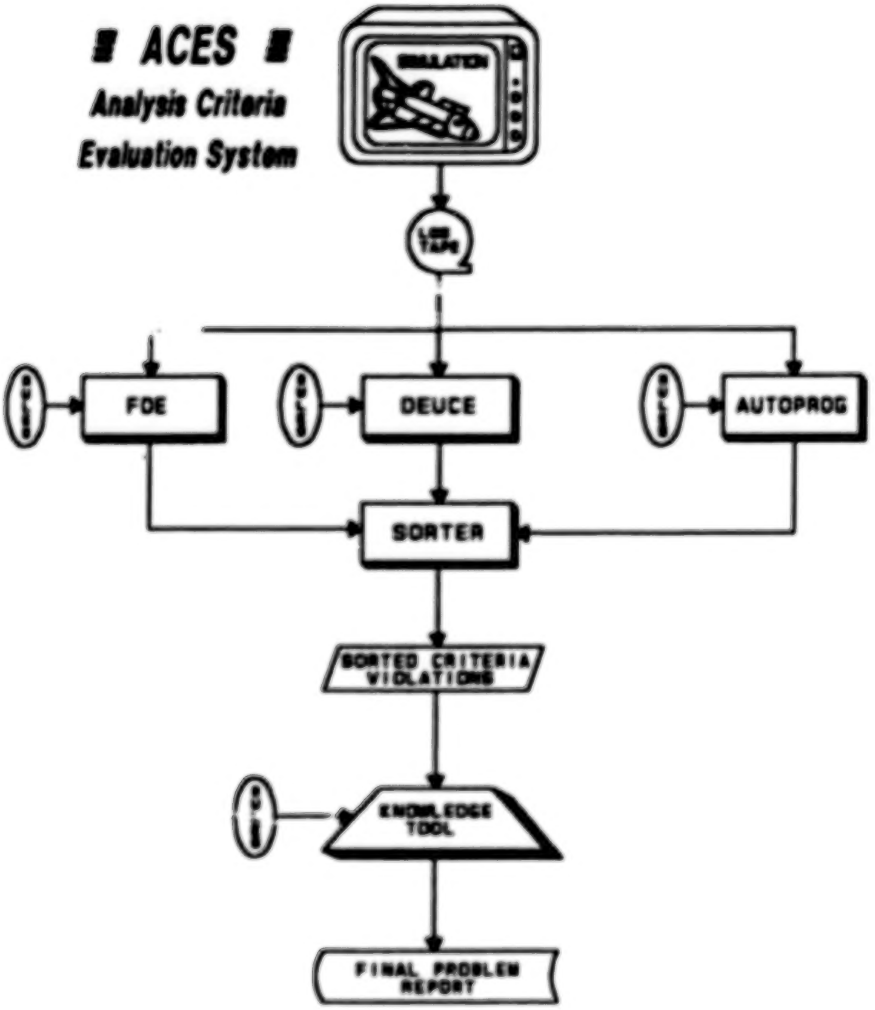
The pass/fail criteria were first prototyped in a mainframe rule-based expert system shell, Expert System Environment (ESE). ESE was not used because of the volume of data to be processed. Instead, three PL/I pre-processors were created to evaluate the pass/fail criteria for the three different types of data to be analyzed.

1. Flight Equipment Interface Device Online Evaluator (FOE) analyzes the sequences of events throughout the trajectory
2. Display Electronics Unit Criteria Evaluator (DEUCE) analyzes the cockpit displays
3. Automation Program (AutoProg) analyzes the plotted parameters

The outputs of the pre-processors are the data items which violated the criteria limits as well as additional data needed for evaluation of the violations. These pre-processors exhibit expert system qualities in that they perform the preliminary analysis that the experts perform.

The initial ACES prototype was enhanced with additional rules to analyze the criteria violations. The methodology for evaluating the pass/fail criteria violations was implemented in another mainframe rule-based expert system shell, KnowledgeTool (KT). KT was chosen because of its speed, modularity and ability to interface with other programs. In ACES, KT runs in batch mode on IBM 30xx MVS processor mainframes. The process of running the pre-processors and expert system for data analysis is automated and no manual intervention is required except to submit the job.

The following diagram illustrates the data flow for ACES. The process begins with the simulation execution. The data is stored on a logtape and then analyzed by the three pre-processors. The results are criteria violations and additional data needed by the KT knowledge base which are sorted and sent to KT. The KT source code is compiled into PL/I executable modules and rule files which are then linked to the KT executable code to run the application. The final results are viewed on the analyst's terminal. After the analyst determines the cause of the problems, the knowledge base can be updated so that it can "learn" the new information which would then be used for future shuttle flights.



SYSTEM VALIDATION/VERIFICATION

Validation of ACES was separated into two phases:

- Pre-processor Tools Verification
- Knowledge Validation

The pre-processor tools were verified using the standard requirements, design and code review techniques. All of the capabilities of each tool were exercised and the results were reviewed by a team including the experts.

The knowledge base rules were validated dynamically by having the experts compare the results produced by ACES to those of manually certified simulations. All differences were resolved and changes were made to the pre-processor tools, the pre-processor rules or to the KnowledgeTool rules. Once the validation was complete for the rules were baselined and put under configuration control.

BENEFITS OF ACES

- Captured expert knowledge
- Reduced total time to certify the flight software
- Increased consistency in data interpretation
- Improved quality of the data analysis
- Reduced dedicated and elapsed time to analyze each simulation

CURRENT STATUS AND FUTURE DEVELOPMENT

ACES is being used in production mode by the Flight Software Certification organization to certify the Shuttle Onboard Flight Software for the Deorbit trajectory. Some of the criteria are still analyzed manually while they are being coded into ACES. For the remaining cases, inputs to AutoProg are being baselined and the knowledge bases are being coded into KT. Research is being conducted to determine how the inference engine and the frames referencing capabilities of KT can be used most effectively.

ACES shows that expert systems can be successfully used to certify/verify complex software systems. Knowledge engineering and acquisition requires considerable time and effort but the retention of the expertise has proved to be very valuable in many areas including analysis consistency and training. Overall, time and resources can be reduced while the quality of analysis is maintained or improved by inserting expert system technology into existing software testing environments.

Test Bed Experiments for Various Telerobotic System Characteristics and Configurations

Neil A. Duffie, Associate Professor
Steven F. Wiker, Assistant Professor
John J. Zik, Associate Researcher

Wisconsin Center for Space Automation and Robotics
University of Wisconsin-Madison

ABSTRACT

Dexterous manipulation and grasping in telerobotic systems depends on the integration of high-performance sensors, displays, actuators and controls into systems in which careful consideration has been given to human perception and tolerance. Research underway at the Wisconsin Center for Space Automation and Robotics (WCSAR) has the objective of enhancing the performance of these systems and their components, and quantifying the effects of the many electrical, mechanical, control, and human factors that affect their performance. This will lead to a fundamental understanding of performance issues which will in turn allow designers to evaluate sensor, actuator, display, and control technologies with respect to generic measures of dexterous performance. As part of this effort, an experimental test bed has been developed which has telerobotic components with exceptionally high fidelity in master/slave operation. A Telerobotic Performance Analysis System has also been developed which allows performance to be determined for various system configurations and electro-mechanical characteristics. Both this performance analysis system and test bed experiments are described in this paper.

INTRODUCTION

Coupling human perceptual and cognitive capabilities to remote electro-mechanical robotic devices shields the human from physical harm. These telerobotic systems permit sustained time on tasks in hazardous or remote environs, reduce transit time to and from the remote site and its associated costs, and reduce or eliminate the engineering and logistic costs of life support systems (e.g. additional design and analysis costs, additional equipment to meet risk-reduction, need for redundant life support equipment, crew life-support and emergency procedure training costs, costs of launching larger payloads, etc.). Telerobotic systems permit the execution of tasks that exceed the performance capacity of fully automated robotic systems, and have demonstrated their worth and are in use in the nuclear industry and in deep-sea exploration and salvage operations. However, the current generation of telerobotic systems have not enjoyed broad commercial success because they are expensive to build and maintain, capable of performing only rudimentary manipulation tasks in a comparatively slow and clumsy manner (i.e. if they can accomplish the task, their performance times range between 8 and 500 times that of human performance), and demand highly trained operators to successfully accomplish assigned tasks. The sensory and perceptual requirements of the task, designed with the human in mind, can overwhelm the telerobot's sensory detection and processing capabilities, and manipulative requirements can exceed the kinematic or positional capacities of the remote manipulator [1].

Assessment of telerobotic system feasibility has been relegated to expensive and time-consuming field trials which often yield performance metrics which are of limited use in evaluating performance potential in dissimilar or alternative tasks. Performance tasks often are not well defined (e.g. manipulator positioning accuracy, force and torque, and operator perceptual requirements are not described), testing methods often are not described in sufficient detail to permit replication and performance comparisons among competitive telerobotic devices, and performance metrics often are of little utility to the engineering community which is interested in application or improvement in telerobotic devices.

Telerobotic devices vary significantly among each other in design and construction. Historically, developers have focused development efforts upon one, or at most a few, telerobotic subsystems using comparatively simple supporting apparatus ensembles to minimize total development time and development costs. For this reason, though the potential number of feasible combinations of alternative telerobotic

subsystems is large, comparatively few implementations have been investigated. Merging a number of promising telemanipulation technologies often requires compromises in engineering design, and ultimately in system performance. The impact of any particular subsystem can be significantly influenced by the nature and performance of interrelated subsystems.

The degree to which an individual subsystem affects overall telerobotic device performance can be determined with accuracy only when considered conjointly with other subsystem designs. Telerobotic devices have been developed with either a specific set of tasks in mind, or a general goal of human capabilities. Once built, a prototype is typically subjected to a set of highly specific operational tests to determine performance feasibility. Regardless of test results, this approach requires that the developer undergo one field test after another to prove that the device is capable when other tasks are considered. Time and expense of field testing impedes marketing capability, and ultimately increases the cost of the device. Moreover, test methods are rarely described in sufficient detail to permit replication or comparison of findings, and performance measures (e.g. successful versus unsuccessful, total completion time, subjective estimates of performance difficulty) are not useful metrics to engineers concerned with efficiently improving the performance capacity of a telerobotic device.

As an example, consider the case of haptic displays. There is little doubt concerning the utility of tactile feedback [2] as exemplified in Figure 1. There are few haptic displays, and no cutaneous display systems which are able to convey a complete sense of touch. Significant research and development efforts have been made in the area of psychophysics [3] (e.g. stimulus perceptual thresholds), and in displays designed to convey alphanumeric characters, or left-right up-down directional cues for vehicle operators. However, little is known about stimulus methods and strategies needed to convey perceptual information [4]. Questions concerning the design of haptic displays are manifold. For example, what stimulus tactor system (i.e. the form of stimulus, tactor size, spatial distribution, tactile and tactor force resolution, etc.) is acceptable given task constraints, mode of stimulation, and necessity of corroborating stimuli (i.e. postural, visual, and auditory feedback) for development of operationally relevant perceptions? Haptic displays must convey information without disrupting perception of master-controller force reflection (i.e. backward masking), keeping in mind operator tolerance and stimulus acceptance issues, and the problem of stimulus adaptation (i.e. requiring greater and greater stimulus intensities to achieve suprathreshold sensations). Significant future efforts will be required in designing haptic displays and assessing their performance in telerobotic systems. This will require test beds in which future displays can be exercised in telerobotic systems, and adequate tools with which to assess their performance and feasibility.

TELEROBOTIC PERFORMANCE ANALYSIS SYSTEM

Comprehensive analytic models, development and testing of

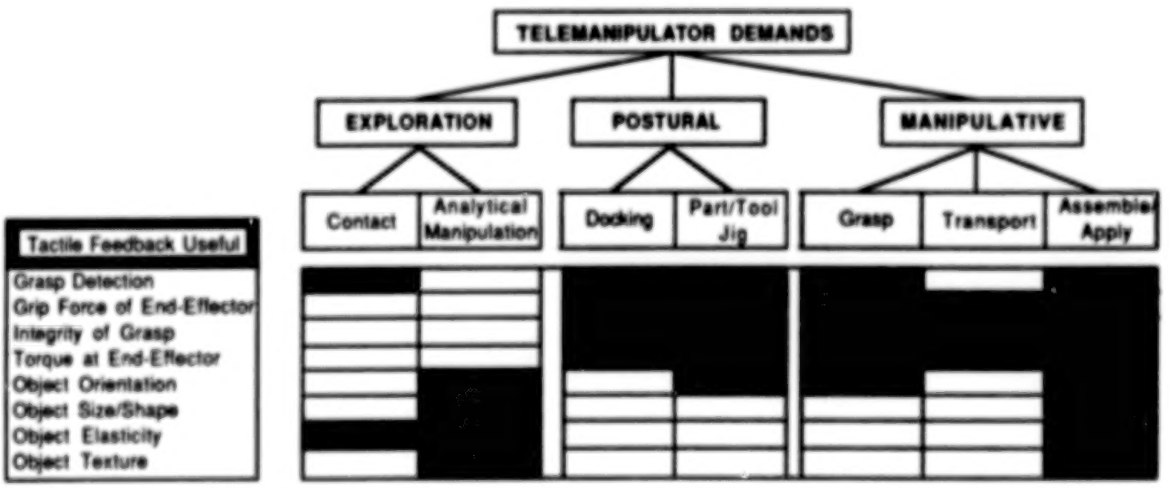


Figure 1. Scope of importance of tactile feedback in telemanipulation

multi-faceted prototypes, and enhancing our knowledge of the effect of interactions between subsystems upon overall system performance can help direct development of capable telerobotic systems that are not overly complex or expensive. In response to this need, WCSAR has undertaken a program to:

a) develop telerobotic work methods analysis procedures;
b) develop terminology used for describing telerobotic performance objectives; and
c) develop performance models and metrics used in describing device performance capabilities.

The industrial community has long accepted this practice, and uses methods engineering models for describing and analyzing human worker and machine performance in manufacturing environments. Using a standardized set of descriptors, task descriptions can be accurately conveyed to other engineers, task descriptions can be entered into computerized performance analysis models, and, thus, systematic comparisons can be made of task performance and cost across telerobotic devices developed within and among laboratories and vendors [5].

Following methods analysis, motor (e.g. Therblig sequence, indexes of difficulty for motor sequences, positioning tolerances, type and force of grasp, etc.), perceptual (e.g. task visual, aural, kinesthetic, and haptic detection demands), and cognitive (e.g. information processing, decision making, etc.) elements of a telerobotic task can be analyzed using a family of telerobotic performance prediction models. In addition to predicting performance feasibility for a telerobotic device of known physical performance characteristics, the models indicate which performance elements which are most troublesome, and what subsystems are most limiting of performance. With this knowledge, an analyst may change the methods of the task, or consider an alternative telerobotic design that is better in the face of performance and cost criteria. Figure 2 graphically shows the organization and process of the Telerobotic Performance Analysis System which is being developed.

Although methods analysis and human performance used in industrial manual assembly operations are well established, new or revised models must be developed for telerobotic systems. Robust models of human performance are based upon intact humans whose perceptual-motor skills are not diminished as they are when coupled to a master-controller and given only limited subsets of sensory information.

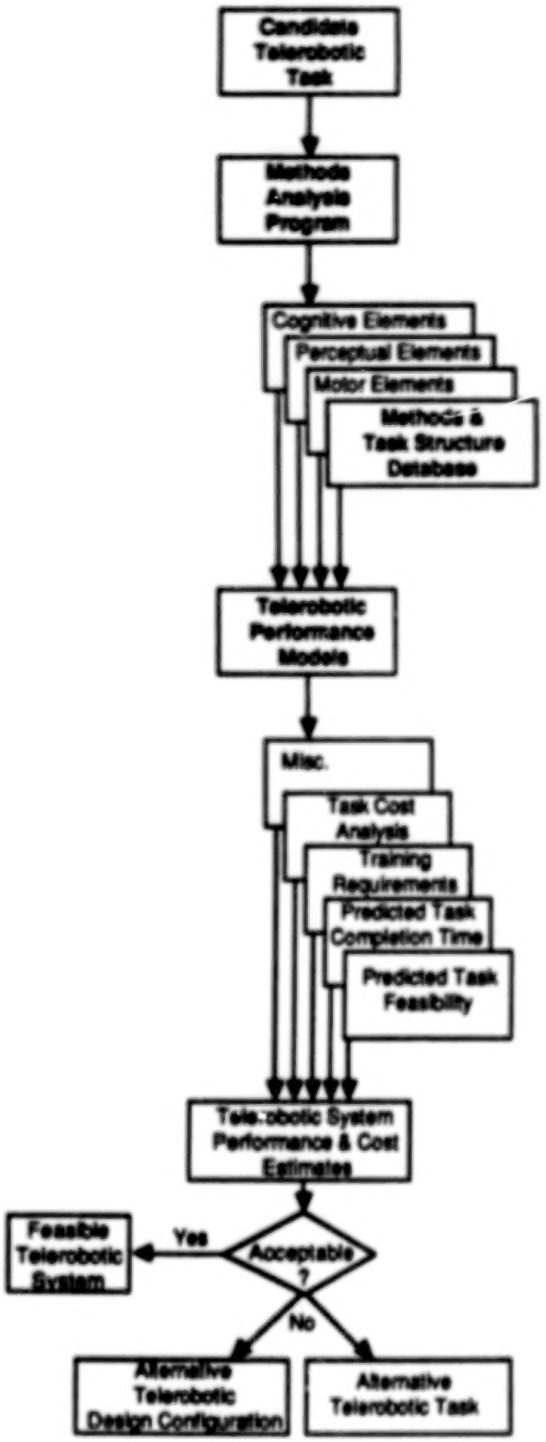


Figure 2. Telerobotics Performance Assessment System

Initially, we will employ motor, perceptual, and cognitive performance models which have demonstrated statistical robustness and operational validity industrial settings. The

goals are to:

a) provide multi-variate design gradients to speed engineering development of telerobotic devices while minimizing data collection;
b) collect data using techniques which provide results which are acceptable to both basic science and engineering communities (i.e. performance findings are scientifically valid, yet metrics have engineering design relevance); and
c) provide on-line guidance to the experimenter regarding the design and implementation of an evolutionary, or "hill-climbing", experimental approach to determining the best mixture of telerobotic subsystems to meet a set of operational objectives.

DEMONSTRATIONS OF TELEROBOTIC PERFORMANCE ANALYSIS

For the purposes of demonstration, the Telerobotics Performance Analysis System will be used to drive the design, and to confirm the performance capabilities, of dexterous telemanipulation systems which provide simple yet compelling perceptions of remote touch. Clearly, there are a number of design variables which must be considered when designing and implementing an integrated end effector, master controller, and haptic display system. This engineering problem is of sufficient challenge to test the ability of the Telerobotics Performance Analysis System to conjointly evaluate several design variables simultaneously, and to expeditiously recommend valid design modifications following limited testing. The first phase of the demonstration will be based on a high fidelity, table-top master/slave gripper in which design variables can be independently modified and controlled. Perceptual-motor performance test findings will be used to direct experimentation and to provide multi-variate design gradients for use in guiding the next phase of the demonstration in which will employ a prototype manipulator (arm and hand), master controller, and haptic display complex in the WCSAR Telerobotics Test Bed. Results obtained will enable the engineering design of future more capable telemanipulator actuation, control, and display subsystems.

Significant advancements have been made in the design and implementation of robotic end effectors. Three-digit and four-digit hand-like "tendon", gear, or direct-driven robotic end effectors have been developed in laboratories concerned with analysis and control of flexible hand-like grasping systems, and actuation and control strategies for multi-articulated grippers. Yet, many fundamental questions concerning end-effector geometry, degrees-of-constraint, actuation bandwidth, actuation and transmission strategies, etc. have not yet been answered satisfactorily. End effectors must resist damage in their operating environment and produce sufficient grasp force, manipulation bandwidth, and grasp compliance or stiffness to meet operational requirements. In addition to these design issues, there is uncertainty about the performance consequence of implementing greater end effector kinematic complexity (e.g. number of articulations within a digit, and number of digits), palmar and volar topology, and sensor integration.

Problems also must be overcome in the design and implementation of a dextrous end-effector master controller [6]. Ideally, the coupling between the controller and the operator's hand should be very stiff for the sake of good position and velocity perception and control. However, stiff coupling schemas result in rapid onset of localized hypoxia, localized muscle fatigue, discomfort, and tremor in the intrinsic muscles of the hand all of which limit operator tolerance and performance capacity. The bulk and limited degrees of freedom of a back-driven master hand-controller are also likely to restrict operator range of motion capability, and ultimately end effector dexterity. Deadspace, backlash, and friction in the master controller and end effector may significantly affect an operator's ability to perform or to recognize small displacements in the end effector.

In order to understand the effects of different forms of sensory feedback, and quantify how the performance of an operator is affected by changes in the electromechanical characteristics of a system, a high-fidelity, single degree-of-freedom, table-top master/slave gripper has been developed by WCSAR. With this system, the ability to test a number of different types of sensors providing high-performance force or tactile feedback to the operator is provided. In conjunction with various forms of sensory feedback, mechanical characteristics of the system such as compliance, mass, friction, backlash, and dynamic bandwidth can be altered, thereby providing a straightforward experimental system which allows Telerobotic Performance Analysis System to be used to quantify performance under various conditions.

The single degree-of-freedom master/slave gripper system was designed to be a nearly ideal electromechanical system. The master and slave devices are identical in design and construction. Figure 3 is a photograph of the system, and Figure 4 shows a layout sketch of one of the devices. Each device consists of two linear DC motors connected in parallel with a stroke of two inches (5 cm). A high-resolution linear encoder is provided for position feedback and velocity estimation. The devices have no backlash, and friction is minimal. Backlash was eliminated by using direct drive actuators and no gear reduction.

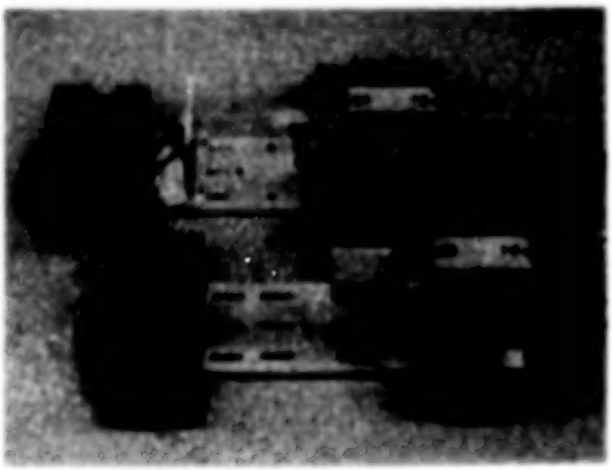


Figure 3. Single degree-of-freedom, table-top master/slave gripper

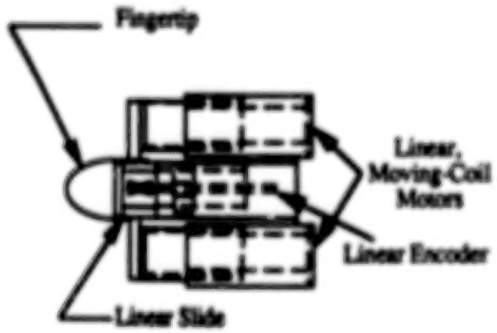


Figure 4. Mechanical detail of slave device (master is structurally identical)

Friction was minimized by using brushless motors and a precision linear slide. The slide is the sole source of mechanical friction with a friction force of less than 0.33 oz. (9.4 g). This is 0.25% of the maximum force which can be generated by the device, and 1/40th of the amount of friction in a typical gripping device. A mounting surface is provided to allow various sensors and displays to be tested with the system. The state-feedback bilateral controller used has active stiffness and damping as shown in Figure 5, and the configuration of the computer control system is shown in Figure 6.

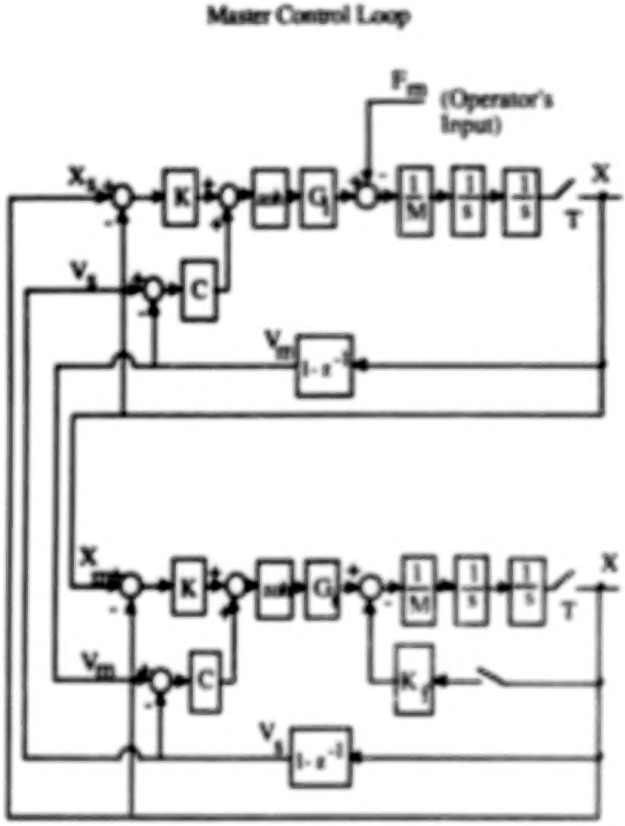


Figure 5. State-feedback bilateral control system with active stiffness and damping

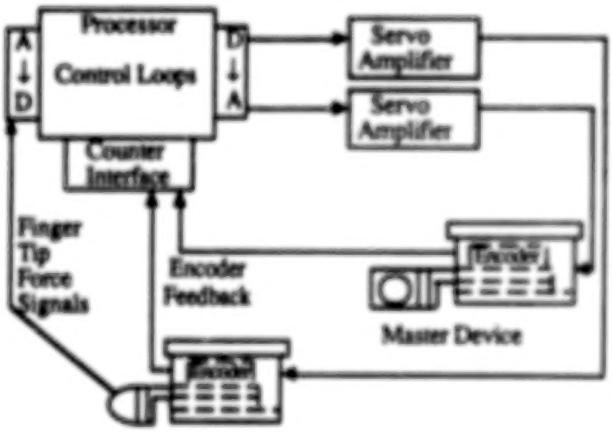


Figure 6. Computer control configuration for master/slave gripper

Initial experiments with this system involve the measurement of an operator's grasp control performance. The system is configured using only the master device as illustrated in Figure 7, and is controlled using the position controller shown in Figure 8. The human subject attempts to maintain a constant force level on the actuator while a multi-frequency sinusoidal position command, X_{com}, is commanded to the actuator. The multi-frequency sinusoidal input is an effective continuous random input acting as a disturbance input to the system. Forces the human subject provides are sensed with a force sensor attached to the mounting surface of the actuator system. The performance measure is the difference between the actual measured forces the human subject applies to the actuator and the reference or mean force level that is intended to be maintained. The parameters that are presently being studied are the stiffness term, K, and the damping term, C. Friction and mass will be studied in the next phase of the experiment, and backlash will then be added in a subsequent phase of experimentation with master/slave operation. The results will be analyzed using the Telerobotic Performance Analysis System, will provide a baseline to determine what the performance tradeoffs are as a function of the above parameters.

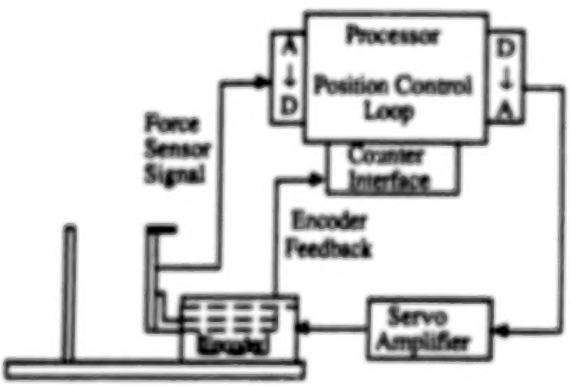


Figure 7. Computer control configuration for initial experiments

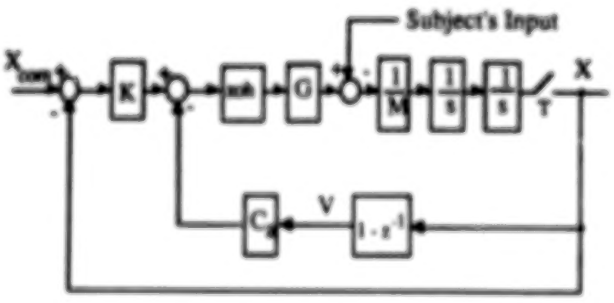


Figure 8. Position control loop for initial experiments

In the present experiment, the system is nearly an ideal linear, second-order system. The stiffness term, K, and the damping term, C, control the placement of the poles in the characteristic equation of the system and can be varied since they are constants in the software of the control system. The system is linear until physical limits are reached. The maximum power limit of the system limits the acceleration of the actuator to 2050 in/sec^2. The maximum natural frequency of the system is limited by a mechanical resonance at 115 Hz. Closed-loop natural frequencies of 35 Hz can be easily obtained, and both natural frequency and damping ratio can be experimentally over a wide range. As and example, a frequency response plot of the system with the natural frequency set at 14 Hz and the damping ratio set at 0.68 is given in Figure 9. Figure 10 is a position versus time plot for a position step input command for the system with this natural frequency and damping ratio.



Figure 9. Position loop frequency response showing magnitude ratio response

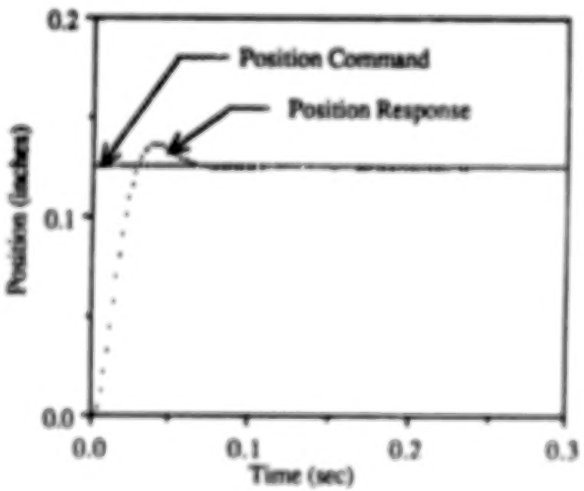


Figure 10. Time response to step input in position command

The goal of these experiments is to establish what physical parameters (i.e stiffness, damping, mass, friction etc.) and characteristics (i.e. types of sensory feedback, haptic display, etc.) are required for a teleoperated system to perform tasks which are characterized by a given an index of difficulty rating [7]. A rating of telerobotic components and technologies based on a task complexity or difficulty index will help to establish least cost approaches to teleoperator development. For example, haptic display technologies assessed using the Telerobotic Performance Analysis System can be rated using results of the form shown in Figure 11.

CONCLUSION

The improvement of dexterous manipulation and grasping capabilities in telerobotic systems will depend on the development and integration of high-performance sensors, displays, actuators, and controls into systems in which careful consideration has been given to human perception and tolerance. One of WCSAR's objectives is the development of these advanced component technologies for use in telerobotic systems for space. As part of this effort, The WCSAR Telerobotics Test Bed [8] has been established in which these technologies can be verified and integrated into telerobotic systems. The layout of the test bed is shown in Figure 12. One of the major systems in the test bed is a telerobotic manipulator with a high-fidelity master/slave hand. The master/slave arm portion of the system consists of a Cincinnati Milacron T^3-726 electric-drive robot and a non-kinematic replica master arm which was designed at WCSAR. The original controller of the robot has been replaced with a new, higher-performance controller designed at WCSAR which is capable of being flexibly programmed in a number of telerobotic operating modes.

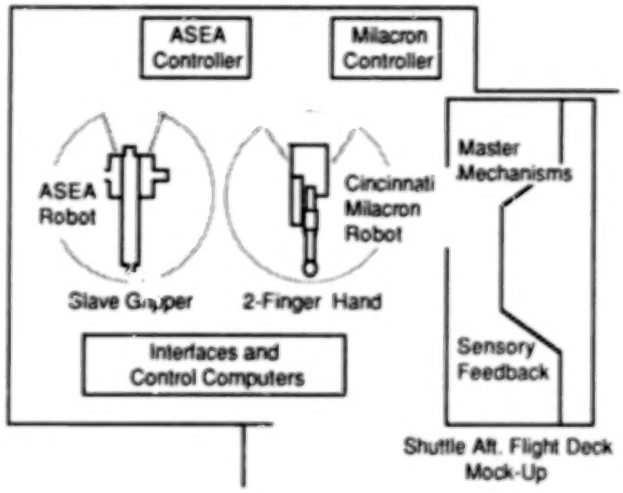


Figure 12. WCSAR Telerobotics Test Bed

The performance of the single degree-of-freedom master/slave gripper described in the previous section has indicated a possible advantage of telemanipulation systems with reduced degrees of freedom but improved electromechanical characteristics and haptic displays over current multiple degree-of-freedom systems. A high-fidelity, two-fingered, master/slave hand therefore has been designed and is currently being tested at WCSAR. The hand consists of a thumb and index finger on the master controller, and a replica of these digits on the slave gripper with corresponding degrees of freedom. The two degrees of freedom are independently controlled by the operator in performing dexterous manipulations. Together with the original arm subsystem, this hand subsystem will allow the assessment of more complex tasks and larger integrated systems with the Telerobotic

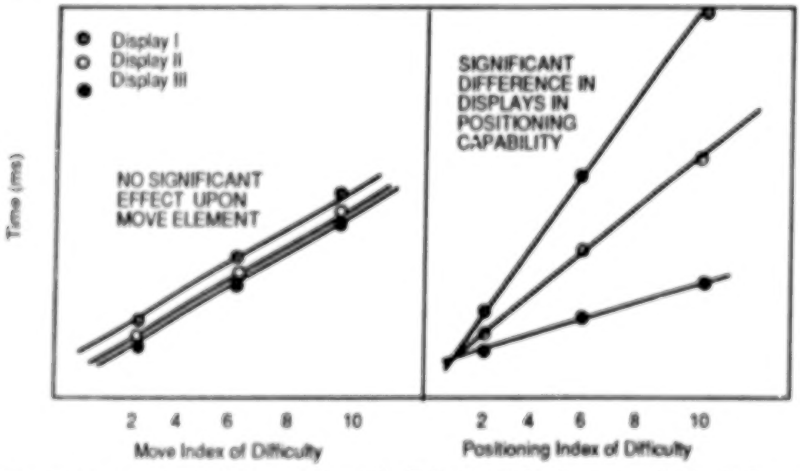


Figure 11. Example of Results from the Telerobotic Performance Analysis System

Performance Analysis System. The hand includes a direct-drive force-reflection system, with minimal friction and zero backlash. The system therefore is capable of supporting high-fidelity telemanipulation with advanced tactile sensors and haptic displays. This combination will be assessed in order to evaluate performance of tasks with high-fidelity telemanipulation and limited degrees of freedom as compared to telemanipulation with many degrees of freedom but low-fidelity.

In conclusion, we are developing the Telerobotic Performance Analysis System to speed engineering development of telerobotic devices, and to provide on-line guidance to the designer in determining the best mixture of telerobotic system components for given operational objectives. Moreover, standardization of telerobotic performance analysis procedures, terminology, performance models, and metrics used in describing device performance capabilities shall assist the scientific and engineering community in its efforts to develop commercially successful telerobotic devices.

ACKNOWLEDGMENTS

This work was supported by the Wisconsin Center for Space Automation and Robotics in part by NASA under grant NAG-W975 and by a consortium of industrial sponsors.

REFERENCES

[1] Hightower, J.K., Smith, D.C., Wiker, S.F., "Development of Remote Presence Technology for Teleoperator Systems", 14th Meeting of the United States-Japan Natural Resources Committe Meeting of the Marine Facilities Panel, Bethesda, MD, Sept 19-20, 1986.

[2] Wiker, S.F., "Tactile-Sensing Techniques Applicable for Telerobots", Technical Doc. No: 1249, Naval Ocean Systems Center, San Diego, CA, January 1988.

[3] Sherrick, C.E. and Craig, J.C., "The Psychophysics of Touch", TACTUAL PERCEPTION: A SOURCEBOOK, W. Schiff and E. Foulke, Ed., Cambridge: Cambridge University Press, 1982, Chapt 2.

[4] Wiker, S.F., "Teletouch Display Development", Technical Doc. No: 1270, Naval Ocean Systems Center, San Diego, CA, 1988.

[5] Kennedy, R.S. et al., "Performance Evaluation Tests for Environment Research (PETER), Report No: 80-R-008, Naval Biodynamics Laboratory, July, 1981.

[6] Wiker, S.F., Hershkowtiz, E., Zik, J.J., "Teleoperator Comfort and Psychometric Stability: Criteria for Limiting Master-Controller Forces of Operatiojn and Feeedback During Telemanipulation", NASA's Conference on Space Telerobotics, Pasadena CA, 1989.

[7] Wiker, S.F., Langollf, G.D., "Shoulder Posture and Localized Muscle Fatigue and Discomfort", Vol. No: 32, Issue No: 20, 1989, Page Nos: 211-237.

[8] Duffie, N., Zik, J., Teeter, R., Crabb, T., "The WCSAR Telerobotics Test Bed", SOAR 88, Dayton, OH, July 20-23, 1988.

NEEDS AND USES OF HUMAN ENGINEERING DATA IN THE DESIGN OF A FORCE REFLECTING EXOSKELETON

Captain M. Jaster
AAMRL/BBA, Wright-Patterson AFB

(Paper not provided by publication date.)

LTM-A-DUAL-ARM REDUNDANT TELEROBOTIC SYSTEM

R. Mixon and W. Hankins III
NASA/Langley Research Center

(Paper not provided by publication date.)

A LABORATORY BREADBOARD SYSTEM FOR DUAL-ARM TELEOPERATION

A. K. Bejczy, Z. Szakaly and W. S. Kim
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109

ABSTRACT

The computing architecture of a novel dual-arm teleoperation system is described in this paper. The novelty of this system is that (i) the master arm is not a replica of the slave arm, it is unspecific to any manipulator and can be used for the control of various robot arms with software modifications, and (ii) the force feedback to the general purpose master arm is derived from force-torque sensor data originating from the slave hand. The computing architecture of this breadboard system is a fully synchronized pipeline with unique methods for data handling, communication and mathematical transformations. The computing system is modular, thus inherently extendable. The local control loops at both sites operate at 1000 Hz rate, and the end-to-end bilateral (force-reflecting) control loop operates at 200 Hz rate, each loop without interpolation. This provides high-fidelity control. This end-to-end system elevates teleoperation to a new level of capabilities via the use of sensors, microprocessors, novel electronics, and real-time graphics displays. The paper concludes with the description of a graphic simulation system connected to the dual-arm teleoperation breadboard system. High-fidelity graphic simulation of telerobot (called Phantom Robot) is used for preview and predictive displays for planning and for real-time control under several seconds communication time delay conditions. High fidelity graphic simulation is obtained by using appropriate calibration techniques.

INTRODUCTION

A laboratory breadboard system has been developed at JPL for dual-arm teleoperation using a novel generalized bilateral control method for robot (or slave) arm control. Generalized bilateral control of robot arms denotes (i) the ability to control the motion of a robot arm from another, dissimilar robot arm or device and (ii) the ability to reflect the forces sensed by the robot hand back to the hand of the operator. Since the controlling device (the hand controller or HC) is not similar to the robot being controlled, the HC can be designed to perform the task of control and force feedback best, and subsequently, this device can be used for the control of different robot arms[1]. To generate force feedback the HC has to be equipped with motors just like a robot and the control electronics of a robot and a HC can be made identical. In space telerobotic applications the control station may be some distance away from the robot so the control computations have to be carried out at two sites, the local or control station site and the remote or robot site.

An evolving electronic system is under development at the Jet Propulsion Laboratory (JPL) that was designed to solve the motor control and computational tasks of generalized bilateral control. This electronic system (The Universal Motor Controller or UMC) was used to build a generalized bilateral robot control system with PUMA 560 manipulators. These manipulators are equipped with Smart End Effectors (SEE) that sense the wrist and the grasping forces. The signals from the SEE are used to achieve force feedback to the hand controller and to achieve shared manual and automatic control of robot arms. An example of this shared control is when during peg insertion into a hole the robot automatically aligns the peg orientation while the operator translates it into the hole.

It is noted that in conventional teleoperation systems the master arm is a one-to-one or scaled replica of the slave arm and force feedback to the master arm is not derived from forces and moments sensed at the robot hand. Instead, it is essentially derived from position error between master and slave arm joints.

Note also that the control and computational system implied in generalized bilateral control of robot arms also forms a natural base for a supervisory control system of telerobots. In a supervisory control system, manual and automatic control can be traded or shared on the level of task space or work space variables. Thus, evolving capabilities in automation can easily be added to the generalized bilateral control and computational system described in this paper.

The breadboard system currently consists of: (1) two six degree-of-freedom (dof) PUMA 560 robot arms, each equipped with a JPL smart robot hand; the hand is a parallel claw device equipped with a six dof force-torque sensor, grasp force sensors and local processing and control electronics. (2) Two six dof generalized Force-Reflecting Hand Controllers (FRHC), each permits one-hand manual commands in six directions, three translation and three orientation commands either in position or in rate mode; the FRHC is unspecific to any manipulator, it can be used for the control of various robot arms with software modifications. (3) Two computing nodes for control and information display, one at the robot site and one at the FRHC (control station) site. (4) A computer graphics terminal at the control station, utilizing (a) a PARALLAX graphics board to generate real-time sensor information displays and (b) an IRIS graphics super workstation to generate real-time perspective images of robot arm motion either on a mono or on a stereo monitor for preview or predictive displays to aid motion planning or control under communication time delay conditions. The current status of the dual-arm teleoperation system with smart hands and with related control station setting is shown in Fig. 1.

In the first part of the paper the electronic architecture and design choices are discussed. This is followed by the description of the current teleoperation system and the upcoming new developments. The last part of the paper contains the description of a graphics simulation system connected to the dual-arm teleoperation breadboard system. High-fidelity graphic simulation of telerobots (called Phantom Robots) is used to create preview and predictive displays for planning and for real-time control of telerobots under several seconds communication time delay conditions. High-fidelity graphics simulation is obtained through appropriate calibration techniques described at the end of this paper.

ELECTRONIC ARCHITECTURE

The UMC architecture has been described in several publications where it can be found in more detail. See [2] and [3]. There are two tasks that have to be performed by such a system.

- Motor control and feedback signal sensing
- Mathematical computations

In our system an integrated approach was used so that both of the above tasks are carried out by a single electronic system. Since the mathematical transformations involved are complex, they cannot be performed by a single processor. This necessitates inter-processor communication and synchronization besides internode communication.

The following are the essential system components for which design choices have to be made:

- Power amplifiers
- Feedback data sensing elements
- Motor control hardware to joint servo processor communication
- Processors
- Inter processor communication
- Inter node communication
- Programming language and development environment
- Motor control algorithm
- Kinematic transformation algorithms

To achieve a compact, integrated package, the power amplifiers and feedback data sensing elements were developed in house. These with the joint processors constitute the UMC and have been described in detail in [2]. In short, this electronics consists of PWM power amplifiers for up to 1 kW motors and provides sensing of motion parameters at servo rates (1000 Hz). Thanks to the NASA technology utilization program, this electronics is now available commercially for up to 10 kW motors either brushed or brushless[4].

The communication from the motor control elements to the joint processor is a private bus called the BLX bus that makes the joint motion parameters memory mapped. It is noteable that with the UMC up to 16 joints can be controlled by a single joint servo processor.

The processor currently used is the NS 32016. There is a large number of processors from which we could choose and the 32000 family has proven to be a very good candidate for our task. The family has a number of processors with a wide performance range and object level compatibility between the members. Its assembly language has proven to be powerful as well as easy to use. The widely used 6800 family would provide less performance, less compatibility between

members and less symmetry in assembly language. Two more advantages of the 32000 family are the relatively small component count and relatively low bus clock rate per unit of performance. The small component count makes it easier to produce a radiation hardened version of a microprocessor, and the relatively slow bus timing makes it possible to time share devices or memory on the bus.

Figure 2 shows the overall architecture of the multibus based distributed computing for our two-node supervisory control system, including the UMC. The main electronic components with the related functions are shown on the board level in Figure 3 (1988 status).

To save development time we used the DB32000 development board which comes with a MULTIBUS interface. This forced us to use MULTIBUS for inter-processor communication. This is a lower bandwidth bus than more recent 32 bit busses. The available bandwidth is, however, more than enough for our application, so the use of MULTIBUS did not hamper the performance of our system. With the upcoming development of new processor boards (still using the 32000 family), a new proprietary bus (the ZBUS) will be introduced that is optimized for high bandwidth shared memory applications.

The inter-node communication currently is performed by a 5 Mbaud fiber-optic link that was developed in-house. Via this fiber optic link a single packet is transmitted every millisecond. This packet carries robot motion commands and also serves as a way of synchronizing all the computations in both the robot and the hand controller nodes. The forward communication link contains a software delay loop to be able to introduce an artificial time delay into the system. This time delay may be set from 0 to 1 seconds in 1 millisecond increments, for time-delay experiments.

Currently the forward packet carries the following information:

- Control mode
- Position change commands for the six degrees of freedom
- Finger grasping force command
- Checksum

Control Modes

The control modes are the following:

- Freeze mode; the robot sets the brake and servos the wrist joints to their positions when freeze mode was entered.
- Neutral mode; the robot is gravity compensated but it may be moved by hand to any position desired. Since the gravity load is compensated by software, when left alone the robot will stay at whatever position it was moved to.
- Current mode; the six bytes following the mode byte will directly command the currents of the six joints. In current mode gravity compensation is still active so at 0 current the robot will not move unless there are external forces acting on it.
- Joint mode; the six motion command bytes will be added to the joint space setpoints, moving the robot in joint space.
- Task mode; the six motion command bytes will be added to the Cartesian setpoints causing robot motion in the Cartesian frame. The so-called task frame is permanently attached to the laboratory, it cannot be redefined.
- Tool mode; the robot is commanded in Cartesian tool frame. This frame is defined by the robot wrist position at the moment the tool mode is activated. This is a Cartesian coordinate system that can be arbitrarily redefined during operation.

If the mode byte of an incoming packet is different from the active mode, the new mode is not entered until 4000 packets come in that all have the same mode bytes. During this intermediate period the robot does not move, any incoming motion bytes are ignored. A new mode has to be active for one second before the robot can be moved in that mode.

For example if the robot is in task mode, the transmitted data carries relative Cartesian coordinates. In every servo loop a change in the range of -D to +D is transmitted, where D is the current speed limit, typically 5 to 10. These changes are added by the receiver to the robot Cartesian setpoint number. This method has a number of merits:

- Small communication bandwidth used
- Error tolerance
- Velocity limiting
- Easy method of indexing the robot

It should be noted that this communication method does not cause any granularity in robot speed whatsoever. It simply limits the granularity of the robot position to 1/10th of a mm. The robot could not be positioned more accurately than that anyway.

The reply packet from the robot side contains the following information:

- Currently active mode

- Wrist forces
- Finger forces
- Finger position
- Joint positions
- Cartesian (task) positions
- Checksum

Development System

The programming language used was the assembly of the 32016 itself since this promised the most performance and the fastest results. It has to be noted that the most convenient development environment such as a C cross compiler and UNIX operating system does not necessarily produce the fastest result and the best program performance. Compilers have the tendency to mask the real world of a processor from the programmer making it harder to generate complex interrupt hierarchies and hardware interfaces. We used a development system that one of us (Szakaly) wrote for the IBM-PC. This system makes it possible to edit and store the assembly source programs in the PC as well as up and download object files. All functions of this development system are integrated so they pass data to each other in the memory of the IBM-AT. If the assembler finds an error for example, it automatically puts the user back into the editor with the cursor on the error. The system also keeps track of the files changed and remembers where each file was modified last. The typical assembly time for a 1000 line program is 15 seconds on a 10 MHz AT which includes the time it takes to write the object output, the symbol table and the memory map files to the disk.

Portions of the teleoperation system such as the force torque display were developed in C using the SYS 32/20 development environment marketed by National Semiconductor.

Control Algorithms

The motor control algorithm is a simple PD control loop. The servo rate is 1000 Hz overall, without interpolation, allowing high gains to be used with the associated high tracking fidelity. The position gains are about 0.1 V/encoder unit. The UMC code generator program is used in the joint level controller. This program assures safe robot control by automatically generating the servo code that controls the joints. There is a set of parameters that have to be specified once for every robot. These parameters are stored in an electrically erasable EEPROM chip. When the program is activated it generates servo code and executes it. There is no possibility of breaking the robot due to human error in the coding.

The code generator is very flexible, it can control any number of motors up to 16, with any combination of hardware elements such as encoders, pots, temperature sensors, motors, brakes. All polarities are menu items so, for example, instead of having to switch the two encoder wires, the user changes the encoder polarity from 'POS' to 'NEG' in the menu. The code generator will use a SUB instruction in place of an ADD in the servo code to accommodate the negative encoder hookup. The motor, the pot, the index and brake polarities can similarly be changed from the menu. The motor control processor interfaces to the rest of the system via the shared memory.

Since the remote node receives Cartesian position setpoints, the inverse kinematic transformation is needed to calculate the robot joint position setpoints. This is carried out by one of the processors on the robot side. This transformation was implemented in integer arithmetic and takes around 700 µsec to execute. Force feedback to the HC is based on robot position error as well as sensor data so the robot end effector Cartesian position has to be computed as well. This is done by computing the robot forward kinematics.

Breadboard Capabilities

As of 6/89 the dual-arm teleoperation system consists of the following major parts (see also Figure 1):

- Two Hand Controller mechanisms
- Local node MULTIBUS cardcages
- Force torque graphic displays
- IRIS workstation with PUMA solid shaded graphic simulation
- IBM-PCs as user interfaces
- PUMA 560 manipulators
- Remote node MULTIBUS cardcages
- Smart End Effectors

The local node cardcage contains the following:

- Two joint interface cards (part of local UMC)
- PWM amplifiers for 8 motors (part of local UMC)
- Joint processor (part of local UMC)
- Kinematic transformation processor
- Communication processor with user interface
- Graphics processor
- Parallax graphics card

The remote node cardcage contains the following:

- Remote node UMC (3 cards and power amplifiers)

- Communication processor
- Smart Hand processor
- Inverse kinematic processor
- Forward kinematic processor

The interfaces are as follows:

- Between cardcages: 5 Mbaud fiber optic links
- From local node to IRIS robot simulation: Fiber optic RS232 at 9600 baud rate.
- From remote node to Smart End Effectors: Fiber optic RS232 at 9600 baud rate for the right hand, fiber optic 3 Mbaud communication for the left hand.

Figure 4 shows the block diagram of the system in its current 1989 status and the interconnections. Figure 5 indicates the timing of events and the sequence of computations. All computations are carried out at a 1000 Hz servo rate. The force feedback signal is currently received at a 125 Hz rate due to the limitation of the RS232 communication channel used. The total round trip time delay is 5 msec for the position error based force feedback and it is around 10 msec for the force-torque sensor based feedback.

The user has a large number of options available through the user interface. Every parameter can be changed on a degree of freedom basis. It is possible to activate a software spring on any degree of freedom that pulls the user's hand back to a center position. Any DOF may be in position or rate mode or it may be turned off. Any degree of freedom can have arbitrary force compliance with a zero or non-zero force setpoint. For example, orientation compliance with zero torque setpoint amounts to automatic peg alignment when performing peg insertion into a hole. An X compliance with non-zero force setpoint will press the end effector against the task board and will maintain contact force. Rate mode is useful when motion over large displacements is desired or when slow, constant velocity motion is the requirement.

The breadboard system multi-mode control flow diagram is shown in Figure 6. The multi-mode control capabilities are described in detail in [5]. Active (that is, force-torque sensor referenced) compliance control and its implementation through a low pass filter is described in detail in [6].

Extensive experiments have been conducted to evaluate the usefulness of these operating modes and force feedback. The data show that force feedback brings an improvement in terms of execution time as well as total force required. The shared control routines also bring about additional improvements. Performance evaluation experiments and results are described in detail in a recent comprehensive report [7].

REAL-TIME GRAPHICS SIMULATION

A real time graphics simulation of the PUMA arm motion has been accomplished by using a Silicon Graphics IRIS-4D GT system. The system is extremely fast both in computation (10 MIPS and 1.1 MFLOPS) and in graphics display. The system can draw 400,000 vectors or 40,000 polygons (4-sided) per second with hidden surface removal and lighting. Thus we could easily achieve the update rate of the PUMA arm graphics simulation to be as fast as the display refresh rate, 60 frames/s for workstation display and 30 frames/s for NTSC video monitor display. Perspective projection was assumed for display, and double buffering was used for the PUMA arm graphics animation to avoid visible flickers or partial drawings. Namely, two display buffers (two 24-bit-per-pixel RGB color image memory buffers) in contrast with a single display buffer were used for display and update in an alternate manner; while one is used for display, the other is used for new drawing, and then the two buffers are switched. Both a solid model with hidden surface removal and a wireframe model with hidden line removal are available for our PUMA arm graphics simulation/animation.

A geometric model of the PUMA 560 arm was constructed by using only 6 object types: 6 boxes, 12 cylinders (frustums), 1 forearm, 1 upperarm, 1 wrist, and 4 fingerhalves. The data structure of the box specifies the box material (color), origin and size. The data structure of the cylinder specifies the cylinder material, origin, bottom and top radii, height, and number of side panels to approximate the side with polygons. The data structure for the other object types were similarly defined. The Denavit-Hartenberg representation was used for the kinematic modeling of the PUMA arm.

Hidden surface removal of the solid model was done by use of the z-buffer of the IRIS graphics system. The z-buffer (24 bits per pixel) contains the z-value data indicating the distance (depth) from the viewpoint for each pixel. At the beginning of each display frame, the z-buffer is initialized to the largest representable z-value (7fffff in hex), while the RGB buffer (24 bits per pixel) containing the red, green, and blue color values is initialized to the background color value. Then during the drawing of polygons, lines, points or characters, the

IRIS graphics system updates the RGB buffer and the z-buffer only for those pixels whose new z-value associated with the current drawing is less than the existing z-buffer value.

The lighting calculations were also done by use of the IRIS graphics system hardware. Once the user defines the material properties (diffuse reflectance, specular reflectance, specular shininess, emission color, ambient reflectance, transparency), light source properties (color of the light source, position or direction of the light source, ambient light associated with the light source), and light model properties (ambient light presented in the scene, scene attenuation factor, local viewer property), the IRIS graphics hardware automatically takes care of the lighting calculations.

It is sometimes advantageous to use a wire-frame model with hidden line removal instead of using a solid model. When the wire-frame model of the PUMA arm is overlaid on the camera view, the viewer can still see the actual camera view of the arm. The wire-frame model with hidden lie removal was accomplished by first drawing the arm with filled polygons of the background color and then drawing the arm again with solid lines of white color. In order to avoid appearance of many broken lines, a small positive depth offset (0.001 in the normalized depth coordinate) was introduced during the filled polygon drawing.

Pop-up menus were provided for the user interface with the PUMA arm graphics simulation. By using a mouse and selecting appropriate menu/submenu commands, the user can perform view control (view angles, view position, zoom), light position control, PUMA arm motion control (6 joint angles and hand opening), screen selection (workstation screen or NTSC video monitor screen), graphics model selection (solid model or wire-frame model), camera calibration, or graphics overlay.

Graphics Overlay on TV Camera Image

The real time graphics overlay of the IRIS graphics output on the video camera image was achieved by using an IRIS video genlock board. The genlock board enables the IRIS graphics output to be synchronized with the incoming video camera signal. It also provides video switching function. Namely, the video output of the genlock board, which is connected to the video monitor for display, can be switched to either the incoming video camera signal or the IRIS graphics output signal, depending upon the alpha-plane value for each pixel. When the alpha-value of the pixel is 255 (ff in hex), the video camera signal is selected for the genlock board video output. When the alpha-value is 0, the IRIS graphics output is selected. Although the major function of the 8-bit alpha-plane of the IRIS graphics system is to allow blending or mixing of two graphics images, in our application we simply used the alpha-plane to control the video switch for the graphics image overlay (or superimposition) on the camera image. During the IRIS graphics rendering, the alpha-values for the background pixels are assigned 255, while the alpha-values for the pixels associated with the PUMA arm are assigned 0. 1 In this way, the PUMA arm graphics model generated by the IRIS graphics system is overlaid on the real camera view. The graphics overlay procedure is schematically summarized in Figure 7.

Camera Calibration

In order to superimpose the PUMA arm graphics model on the camera view of the actual arm, camera calibration is necessary. In our implementation, camera calibration was achieved by an interactive cooperation between the human operator and the system [8]. The operator provides the correspondences between object model points and camera image points by using a mouse. Thereafter the system computes the camera calibration matrix. The calibration procedure is summarized in Figure 8.

As the human operator selects the data entry mode from the camera calibration menu, the PUMA arm graphics model is overlaid on the real camera view, both the model and the actual camera view appearing on the video monitor screen (Fig. 9). At this stage, the graphics model view and the camera view are not aligned. In fact, the human operator is allowed to change the viewing condition (view angle, view position, zoom) of the model arm at any time during this data entry mode, so that the human operator can find and indicate corresponding points easily. Thirty three vertices (corner points) of the PUMA arm model were pre-selected as object points for camera calibration. As seen in Figure 9, these object points are indicated by square marks on the model arm. For clarity, only visible object points are marked.

The operator first picks an object point by clicking the square with a mouse. When the square is successfully picked, the unfilled square is changed to a filled square. The "pick" function call of the IRIS graphics system is efficiently used to identify which object

point is actually picked. After the identification, the 3-D position of the object point is directly obtained from the geometric model of the PUMA arm. This picking process enables us to determine the 3-D position of the object point, even though a mouse click gives only 2-D screen coordinates. After having picked an object point, the operator indicates, on the camera view of the arm, the location of the corresponding image point by clicking a mouse. This picking-and-clicking procedure is repeated until all desired object points and their corresponding image locations are entered. The data entered are now used to compute the camera calibration matrix.

The 4x3 camera calibration matrix describes the relation between 3-D object points and their corresponding 2-D image points by using homogeneous coordinates. With the assumption that the camera view can be modeled by an ideal pinhole camera model as a perspective projection of the 3-D world onto the 2-D image plane, we can consider the camera calibration matrix M as being composed of several simple transformations. While it is possible to decompose the matrix in a variety of ways, the particular decomposition chosen is as follows:

M = (rotate)(translate)(project)
(scale)(crop) = (3-D viewing transform)(perspective projection)
(2-D viewport transform)

The viewing transformation transforms object coordinates (x,y,z) to camera viewing coordinates (x_v,y_v,z_v) by a rotation and translation. The perspective projection transforms the viewing coordinates to image-plane coordinates (u,v). The viewport transformation (window-to-viewport mapping) maps image-plane coordinates to actual screen coordinates (u_s,v_s) by scaling and cropping (translation of the image center) within the 2-D image plane.

$$u_s = s_x u + c_x,$$
$$v_s = s_y v + c_y$$

There is a standard linear least-squares method that can compute the camera calibration matrix M, when 6 or more object points and their corresponding images are given [9], [10]. Once M is obtained, we can recover both intrinsic (2-D image scaling and cropping parameters including camera focal length) and extrinsic (camera position and orientation) camera parameters [11], [12]. However, our testings indicate that recovering camera parameters by this technique, especially scaling and cropping parameters, is very sensitive to measurement errors.

Fortunately, in or application the camera scaling and cropping parameters can be defined to be identical to the graphics viewport parameters. The full size of the camera view displayed on the video monitor screen is normally equal to the full size of the IRIS graphics output in NTSC mode displayed on the same screen since these two are synchronized by the IRIS genlock board. Thus the scaling and cropping parameters of the camera view are assumed to be identical to the graphics viewport parameters. In the NTSC mode of the IRIS graphics system, the screen size is defined as (XMIN,XMAX, YMIN,YMAX) = (0,645,0,484). Thus, viewport transformation parameters are given by $s_x = c_x = XMAX/2$ and $s_y = c_y = YMAX/2$, and so are the camera scaling and cropping parameters. Thus, instead of computing the camera calibration matrix M, we first transform (u_s,v_s) screen coordinates to (u,v) image-plane coordinates for each image point by

$$u = (u_s - c_x)/s_x,$$
$$v = (v_s - c_y)/s_y.$$

Then, we compute the camera calibration matrix C that relates 3-D object coordinates (x,y,z) and 2-D image-plane coordinates (u,v) without 2-D image scaling and cropping.

C = (rotate)(translate)(project)

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_1 & t_2 & t_3 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

where f is the camera focal length.

A linear least-squares method can be used to determine the 12 elements of the 4x3 camera calibration matrix C, when 6 or more object points and their corresponding images are given [9],[10]. However, the linear method does not guarantee the orthonormality of the rotation matrix. In our graphics overlay application, the orthonormalized rotation matrix may be preferred. Orthonormalization can be applied after the linear method, but this does not yield the least squares solution. In general, a nonlinear least-squares method has to be employed if we wish to obtain the solution that satisfies the orthonormality of the rotation matrix.

In the nonlinear method, instead of using 9 elements of a rotation matrix, three angles (pan, tilt, swing) are used to represent the rotation. In our current design, all three camera calibration algorithms are available: (i) a linear least-squares method, (ii) orthonormalization after the linear method, (iii) a

nonlinear least-squares method. The algorithms above can be used for both cases: when the camera focal length f is given and when f is unknown. The solutions of the camera calibration matrix C obtained by the above algorithms are stored in different files. The user can pick any one of the camera calibration matrix solutions for rendering the PUMA arm graphics model and superimposing on the camera view.

The PUMA arm graphics model superimposed on the actual camera view after the camera calibration is shown in Figure 10 for the surface model and in Figure 11 for the wire-frame model. Also indicated on these figures is the predictive display "phantom robot" effect under communication time delay condition. As seen on the right side of Figures 10 and 11, the graphics robot image (the "phantom robot") has moved off from the real robot image on the screen to a location commanded by the operator. When the "phantom robot" motion on the screen is controlled by the operator in real time then the operator can see that motion against the real environment on the screen in real time, provided that the environment on the screen is a static one. The real robot image will follow the motion of the "phantom robot" graphics image after some communication time delay and will stop at the location of the "phantom robot" image on the screen, provided that the geometric calibration of the "phantom robot" graphics image relative to the real robot image on the screen was performed correctly before the motion started.

CONCLUSION AND FUTURE PLANS

The main conclusion is that this end-to-end dual-arm breadboard system elevates teleoperation to a new level of capabilities via the use of sensors, microprocessors, novel electronics, and real-time graphics displays. The new control and task performance capabilities have been verified and evaluated for single-arm operation through a statistically significant set of control experiments as reported in [7]. Dual-arm task performance experiments and time-delayed control experiments using predictive display graphics image of robot arm ("phantom robot") will be carried out in the near future.

Future plans in control system and electronics development affects the following areas:

- Processors and bus architecture
- Communication
- Smart end effectors
- Software development environment
- Supervisory control software

The upcoming new devices are the following:

A new processor card containing two of the NS 32016 processors using the new advanced bus interface and 5 Mbit fiber optic links. This processor card can also be used for upcoming flight experiments.

Another processor card using the NS 32332, the new advanced bus interface, 5 Mbit and 15 Mbyte fiber optic links.

A new smart hand featuring very high (10 kHz) data rates with a 12 bit A/D and the new fiber optic link. The actual servo rate will be limited by the host processor to about 5 kHz, this data will be processed to the 1 kHz rate of the rest of the system as described in [13].

After some experience with the new assembler, improvements will be made to the syntax such that the usage will have the appearance of a high level language. This will provide many of the benefits of high level languages without the associated performance and control loss.

When the new hardware is available, the control software will be upgraded to include evolving supervisory control capabilities in model- and sensor-referenced automatic control of the dual-arm system.

The plans also include the upgrade of the dual and non-redundant (six d.o.f.) arm hardware to a dual and redundant (eight d.o.f.) arm system.

Future plans in real-time computer graphics development include (i) the use of computer controlled TV cameras and (ii) graphics overlays of object models on the TV image. Use of computer controlled TV cameras will provide the capability of using a single complete camera calibration for a task scenario since the camera parameters will automatically be known for all different settings of camera position, orientation and zoom. Graphics overlays of object models on the TV image will enable preview/predictive simulation of sensor-referenced control.

ACKNOWLEDGMENTS

Control electronics and software was developed by Z. Szakaly, and the electronics hardware was built by E. Barlow. Graphics image development was done by S. Venema, and graphics overlay calibration techniques were developed by W.S. Kim.

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

REFERENCES

[1] Bejczy, A.K. Salisbury, J.K. Jr., "Controlling Remote Manipulators Through Kinesthetic Coupling." Computers in Mechanical Engineering (CIME) Vol. 2, No. 1, July 1983, pp. 48-60.

[2] Bejczy, A.K. Szakaly, Z.F., "A Synchronized Computational Architecture for Generalized Bilateral Control of Robot Arms," Proceedings of the Conference on Advances in Intelligent Robotic systems, by SPIE and the International Society for Optical Engineering Cambridge, MA, Nov. 1-6, 1987.

[3] Bejczy, A.K. Szakaly, Z.F., "Universal Computer control System (UCCS) for Space Telerobots," Proceedings of the IEEE International Conference on Robotics and Automation, Raleigh, NC, March 30-Apr. 3, 1987, pp. 318-324.

[4] Motion Tek, Box 9, Lord Ave., Brunswick, NY 12180.

[5] Bejczy, A.K., Hannaford, B., Szakaly, Z.F., "Multi-Mode Manual Control in Telerobotics," Proceedings of ROMANSY'88, Udine, Italy, Sept. 12-15, 1988.

[6] Szakaly, Z.F., Kim, W.S., Bejczy, A.K., "Force-Reflecting Teleoperated System with Shared and Compliant Control Capabilities," Proceedings of NASA Conference on Space Telerobotics, Pasadena, CA, Jan. 31-Feb. 2, 1989.

[7] Hannaford, B., Wood, L., Guggisberg, B., McAffee, D., Zak, H., "Performance Evaluation of a Six-Axis Generalized Force-Reflecting Teleoperator," JPL Publication 89-18, June 15, 1989.

[8] Kim, W.S., and Stark, L., "Cooperative Control of Visual Displays for Telemanipulation", Proc. IEEE Int. Conf. on Robotics and Automation, pp. 1327-1332, Scottsdale, AZ, 1989.

[9] Sutherland, I.E., "Three-Dimensional Data Input by Tablet", Proc. IEEE, vol. 62, no. 4, pp. 453-461, 1974.

[10] Ballard, D.H., and Brown, C.M., Computer Vision, Prentice-Hall, 1982.

[11] Ganapathy, S., "Decomposition of Transformation Matrices for Robot Vision", IEEE Int. Conf. oin Robotics and Automation, pp. 130-139, 1984.

[12] Strat, T.M., "Recovering the camera parameters from a transformation matrix", Proc. DARPA Image Understanding Workshop, pp. 264-271, 1984.

[13] Bejczy, A.K., Szakaly, Z., Ohm, T., "Impact of End Effector Technology on Telemanipulation Performance," - see elsewhere in this Proceedings.

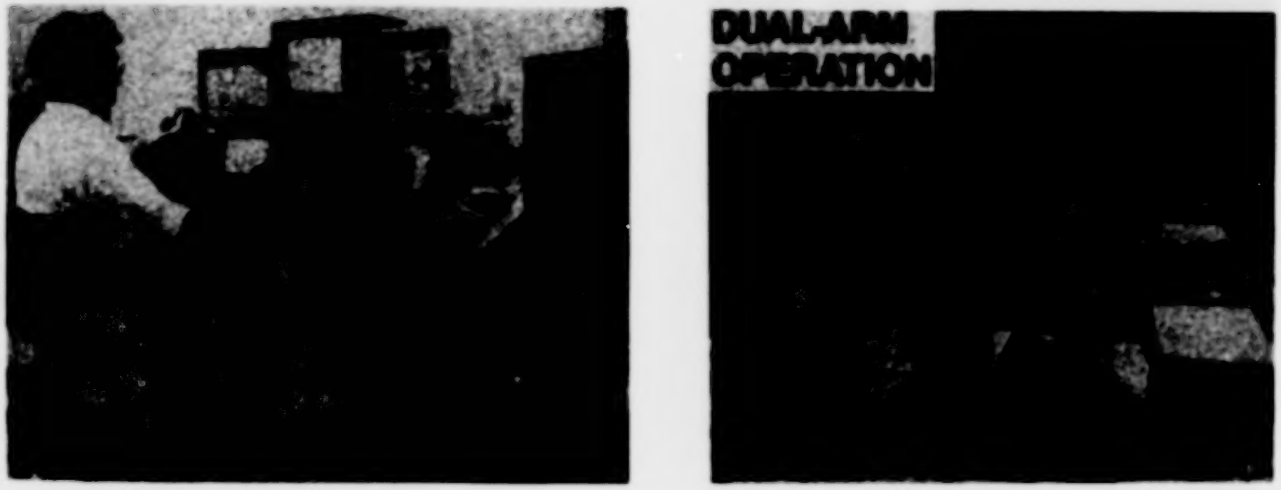


Figure 1. Laboratory Breadboard System for Advanced Dual-Arm Teleoperation (1989)

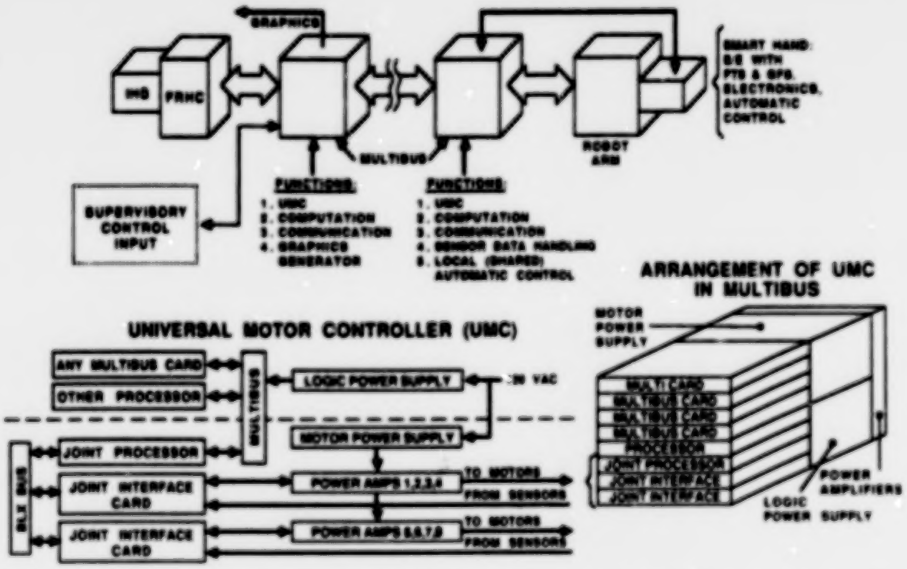


Figure 2. Electronics Architecture of Distributed Two-Node Supervisory Control System

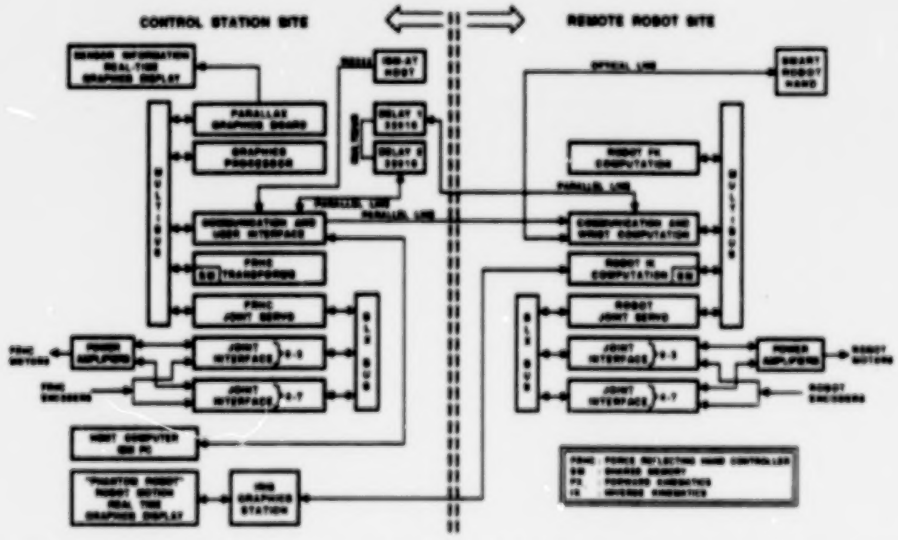


Figure 3. Board-Level Components of Distributed Two-Node Supervisory Control System Electronics (1988)

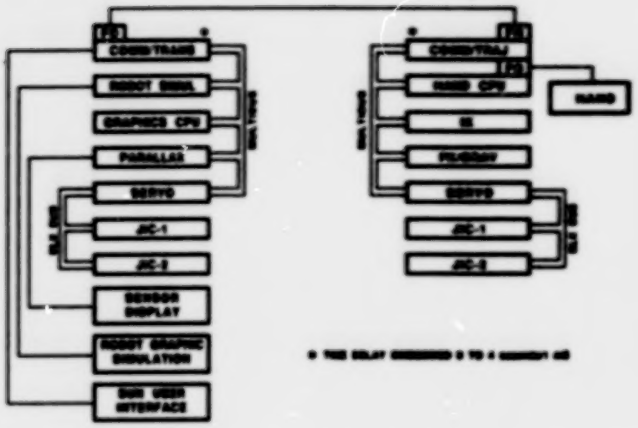


Figure 4. Supervisory Control System Electronics Upgrades (1989)

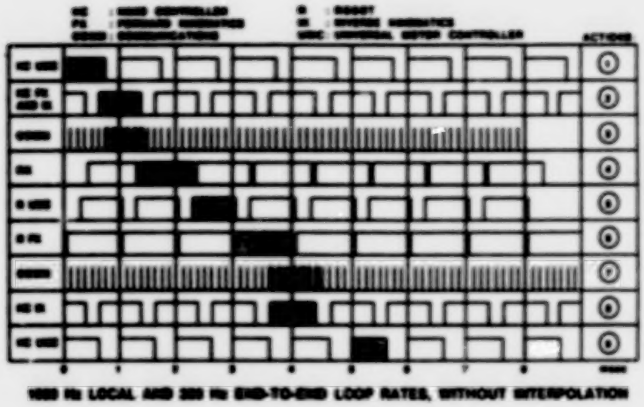


Figure 5. Bilateral Control Communication Timing Diagram

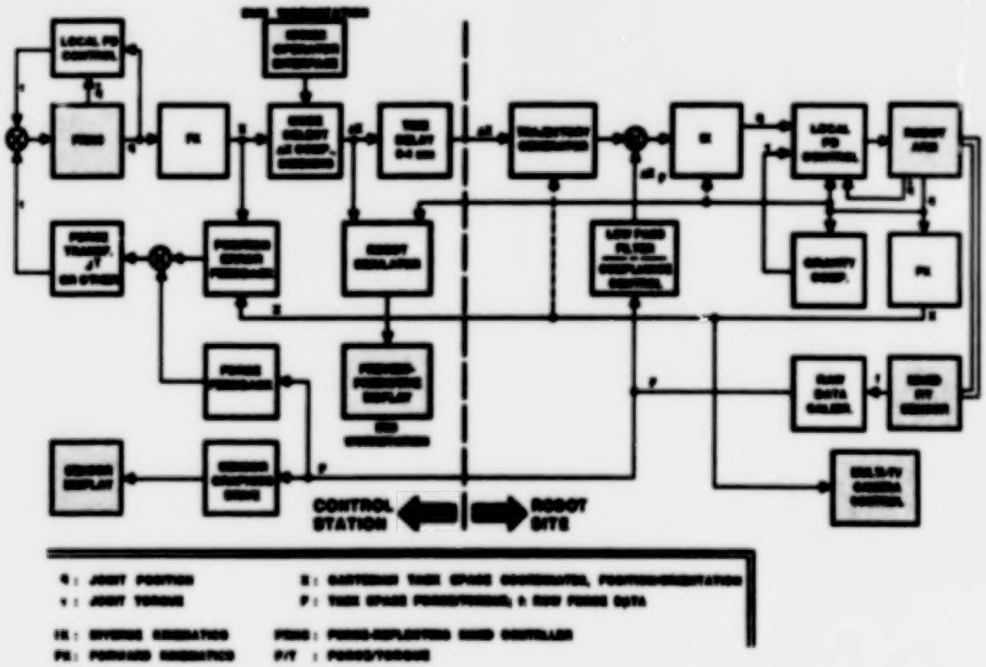


Figure 6. Advanced Teleoperation Control System Block Diagram

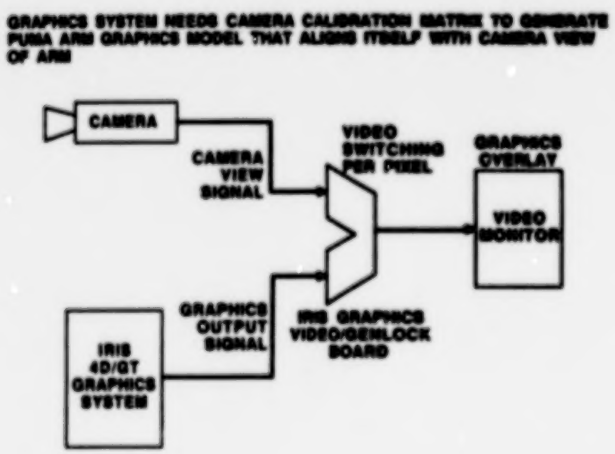


Figure 7. Graphics Video Overlay Procedure

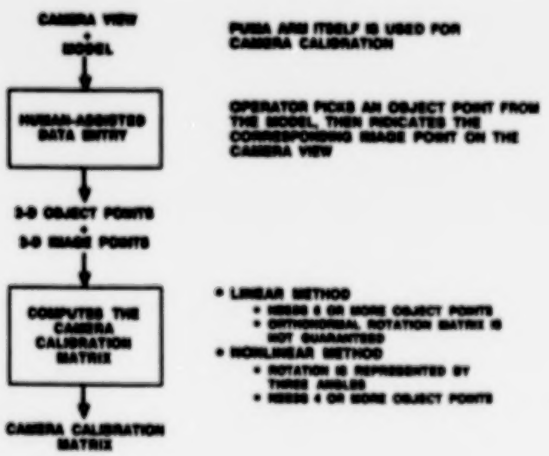


Figure 8. Graphics Calibration Procedure

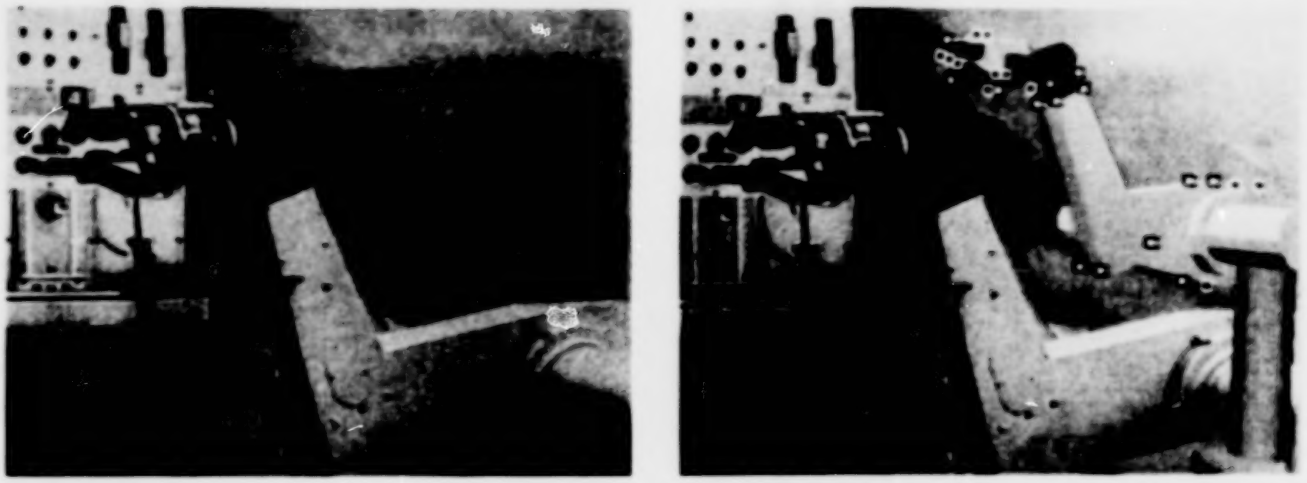


Figure 9. Visual/Manual Calibration of Graphics Overlay

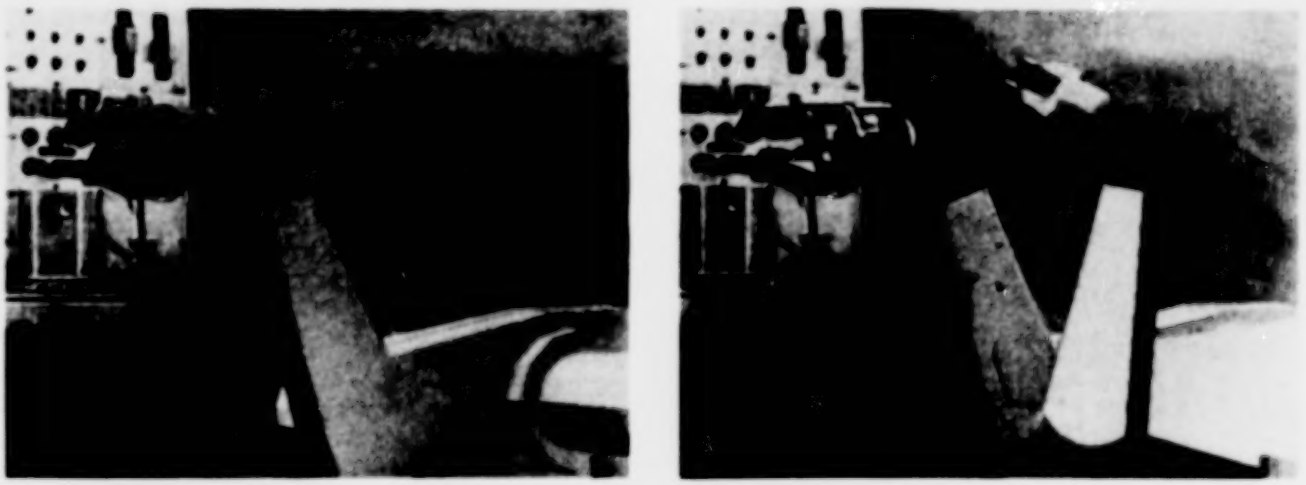


Figure 10. Solid Shaded Polygon Graphics "Phantom Robot" Calibrated Overlay and Predictive Display

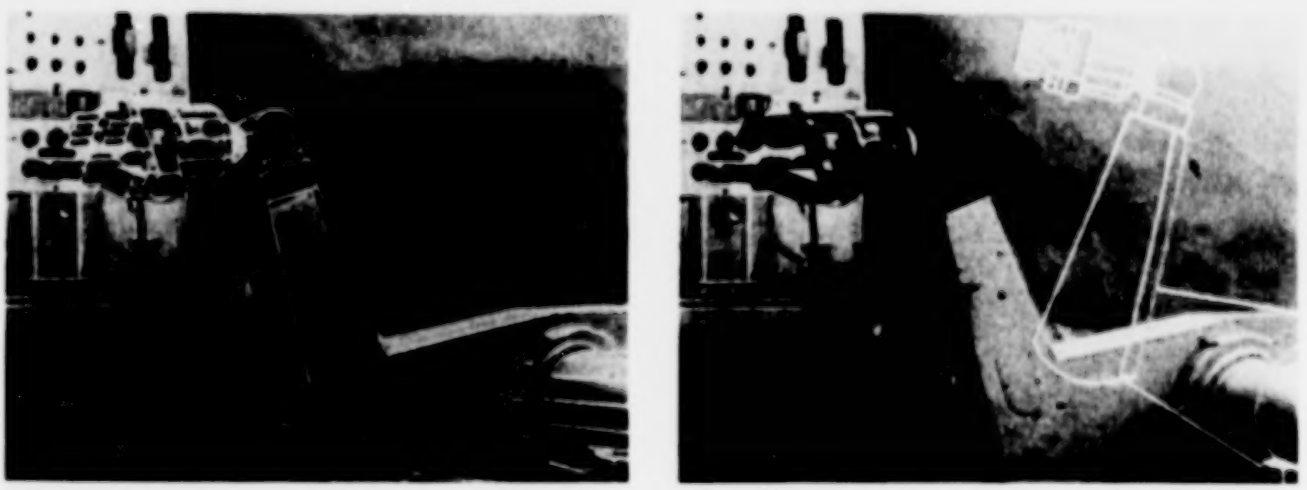


Figure 11. Wire-Frame Graphics "Phantom Robot" Calibrated Overlay and Predictive Display

PROCESS AND REPRESENTATION IN GRAPHICAL DISPLAYS

Douglas J. Gillan
Lockheed-ESC
P.O. Box 58561
Houston, TX 77258

Robert Lewis
Rice University
Psychology Department
Houston, TX

Marianne Rudisill
NASA/JSC
SP34
Houston, TX

ABSTRACT

This paper examines how people comprehend graphics. Graphical comprehension involves the cognitive representation of information from a graphic display and the processing strategies that people apply to answer questions about graphics. Research on representation has examined both the features present in a graphic display and the cognitive representation of the graphic. The key features include the physical components of a graph, the relation between the figure and its axes, and the information in the graph. Tests of people's memory for graphs indicate that both the physical and informational aspects of a graph are important in the cognitive representation of a graph. However, the physical (or perceptual) features overshadow the information to a large degree.

Processing strategies also involve a perception-information distinction. In order to answer simple questions (e.g., determining the value of a variable, comparing several variables, and determining the mean of a set of variables), people switch between two information processing strategies: (1) an arithmetic, look-up strategy in which they use a graph much like a table, looking up values and performing arithmetic calculations, and (2) a perceptual strategy in which they use the spatial characteristic of the graph to make comparisons and estimations. The user's choice of strategies depends on the task and the characteristics of the graph. The paper concludes with a theory of graphic comprehension.

INTRODUCTION

To survive and succeed in the world, people have to comprehend both diverse natural sources of information, such as landscapes, weather conditions, and animal sounds, and human-created information artifacts such as, pictorial representations (i.e., graphics) and text. Researchers have developed theories and models that describe how people comprehend text (for example, see [8]), but have largely ignored graphics. However, an increasing amount of information is provided to people by means of graphics, as can be seen in any newspaper or news magazine, on television programs, in scientific journals, and, especially, on computer displays.

Our initial model of graphic comprehension has focused on statistical graphs for three reasons: (1) recent work by statisticians which provides guidelines for producing statistical graphs (Bertin [2], Cleveland and McGill [4, 5], and Tufte [10]) could be translated into preliminary versions of comprehension models; (2) statistical graphs play an important role in two key areas of the human-computer interface -- direct manipulation interfaces (see [7] for a review), and task-specific tools for presenting information, e.g., statistical graphics packages; and (3) computer-displayed graphs will be crucial for a variety of tasks for the Space Station *Freedom* and future advanced spacecraft. Like other models of human-computer interaction (see [3], for example), models of graphical comprehension can be used by human-computer interface designers and developers to create interfaces that present information in an efficient and usable manner.

Our investigation of graph comprehension addresses two primary questions -- how do people represent the information contained in a data graph and how do they process information from the graph? The topics of focus for graphic representation concern the features into which people decompose a graph and the representation of the graph in memory. The issue of processing can be further analyzed as two questions, what overall processing strategies do people use and what are the specific processing skills required?

GRAPHIC REPRESENTATION: FEATURES OF GRAPHIC DISPLAYS

Both Bertin [2] and Tufte [10] address the features underlying the perception and use of graphs. Bertin [2] focuses on three constructs, (1) "implantation", i.e., the variation in the spatial dimensions of the graphic plane as a point, line, or area; (2) "elevation", i.e., variation in the graphical element's qualities -- size, value, texture, color, orientation, or shape; and (3) "imposition", i.e., how information is represented, as in a statistical graph, a network, a geographic map, or a symbol. Tufte [10] proposes two features as important for graphic construction, data-ink and data density. Tufte describes data-ink as "the non-erasable core of a graphic" [10, p.93] and provides a measure, the data-ink ratio, which is the "proportion of a graphic's ink devoted to the non-redundant display of data-information" [10, p.93]. Data

density is the ratio of the number of data points and the area of the graphic. Tufte's guidelines call for maximizing both the data-ink ratio and, within reason, the data density, in other words, displaying graphs with as much information and as little ink as possible.

Both Bertin's and Tufte's ideas about the features of data graphs were derived from their experience as statisticians, rather than from experimental evidence. We decided to fill the empirical void concerning the features underlying graphic comprehension. In our first experiment, people simply judged the similarity in appearance and information displayed by all possible pairs of 17 different types of graphs (that is, 136 pairs of graphs). The graphs ranged from the familiar (line graphs, bar graphs, and scatter plots) to the more unusual (star graphs, ray graphs, and stick man graphs). The similarity judgments were analyzed with multivariate statistical techniques, including (1) cluster analysis, which shows the groupings or categories (clusters) that underlie people's judgments about a set of objects, and (2) multidimensional scaling (MDS), which shows the linear dimensions underlying people's similarity judgments. The logic of these analyses was that people would cluster graphs and place graphs along dimensions based on the features of the graph [9].

The cluster analyses indicated that people group graphs, at least in part, according to the physical elements of the graphs. Key clusters included graphs in which points were the dominant element (the two types of scatter plots -- the range and density graphs), graphs consisting of angular lines (the pie, ray, stick man, 3-dimensional, and star graphs) and graphs consisting of straight lines (the surface, textured surface, and stacked bar graph), and those consisting of solid areas (the column and bar graphs). The categorization of the graphs according to physical elements agrees generally with Bertin's [2] construct of implantation.

The MDS analyses of the similarity judgments were combined with a factor analysis which resulted in three factors, each consisting of one informational dimension and one perceptual dimension, which accounted for 97% of the data. One factor differentiated perceptually simple graphs (e.g., the bar and line graphs) from perceptually complex graphs (the scatter plots, the 3-dimensional graph, and the surface graphs). A second factor separated graphs for which axes were unnecessary to read the graph (the pie, star, 3-dimensional, and stick man graphs) from those for which the axis contained information (especially the modified scatter plots -- the range and density graphs [10]). Finally, the third factor tended to have informationally complex graphs (those with the most data) at one end and informationally simple graphs (those with the least data) at the other end. Accordingly, we hypothesize that people decompose a graph according to its perceptual complexity, figure-to-axes relation, and informational complexity. A subsequent experiment has shown that each of these three factors relates to people's speed and accuracy in answering questions using these graphs [6].

GRAPHIC REPRESENTATION: REPRESENTATION IN MEMORY

The previous section of this paper addressed the features present when a user looks at a graphic. This section addresses the features that the user walks away with. Accordingly, the experiments looked at how a user represents the information from a graphic in memory.

Our research on memorial representation of graphics involved a simple experimental design: Our subjects worked with a set of graphs on one day, then we assessed what they retained about the graphic on a second day. The initial, training day consisted of one trial with each of six different graphs during a 30 second trial. For 3 graphs, the subjects answered questions about the graphs (e.g., What is the mean of the variables in the graph? and Which is has the greater value, variable A or variable B?). For the other 3 graphs, they identified and drew the perceptual components of the graph, each component in a separate box. (For example, in a line graph a subject might draw the points representing each variable, the lines connecting the points, the axes, verbal labels, and numerical labels.)

Twenty-four hours after training, we tested the subjects using two different methods. We gave one group of 16 subjects a recognition test in which they looked at 24 different graphs and had to say whether they had seen precisely that graph during the training session. We constructed the 24 test graphs isystematically. Each of the 6 graphs from the training session were presented during the test. Each training graph had 3 "offspring" that served as the distractors (or incorrect test stimuli) during the test. One type of distractor contained the same data as the training stimulus, but used a different graph type to display the data (New Graph-Same Data); a second distractor displayed the data using the same type of graph, but had different data from the training graph (Same Graph-New Data); the third distractor differed from the training graph in both graph type and data (New Graph-New Data). Perfect recognition would have resulted in 100% yes answers to the training graphs and 0% yes answers to the distractors. A second group of 14 subjects received a recall test in which they were asked to draw the graphs from Day 1 in as much detail as they could remember.

The results showed that people's recognition of the training graphs was very good. They correctly recognized the training graph 88% of the time, with little difference between the graphs used during training in the perceptual task (85% recognition) and those used in the informational task (90% recognition). Although false recognitions of the distractors were low overall (10% yes answers to distractors), the distribution of false recognitions was interesting. Of the 39 false recognitions by the 16 subjects, 29 (74%) were made to the Same Graph-New Data distractor, Friedman test chi-square (2 df) = 10.1, $p<.05$. The high false recognition rate when the same graph type was used (30% false recognitions to that distractor) suggests that the perceptual type of the graph has a strong

representation in memory. We found that both training with an informational task and training with a perceptual task yielded similar high proportions of the total false recognitions for the Same Graph-New Data distractor -- 77% and 70%, respectively.

The results from the recall test provide even greater support for the hypothesis that the representation of the graph type and certain perceptual features was exceptionally strong. Subjects had good recall for the graph type (71% of the graphs), the presence or absence of axes (71% correct recall of axes), and the perceptual elements (lines, areas, and points) in the graphs (53% correct recall of graph elements). In contrast, recall of information from the graphs was generally poor. For example, subjects had low recall rates for the number of data points in the graph (29% correct recall), the quantitative labels on the axes (10% of the labels), and the verbal labels of the axes and data points (12% of verbal labels). They recalled the correct spatial relations between data points only 22% of the time. In addition to showing the strength of the perceptual representation, these data suggest that the perceptual and informational representations of a graph are independent.

STRATEGIES FOR PROCESSING INFORMATION IN A GRAPHIC

Based on formal thinking aloud protocols, as well as informal discussions with users, we have hypothesized that people use two different types of strategies when processing information from a data graph -- an arithmetic, look-up strategy and a perceptual, spatial strategy. With the arithmetic strategy, a user treats a graph in much the same way as a table: using the graph to locate variables and look up their values, then performing the required arithmetic manipulations on those variables. In contrast, the perceptual strategy makes use of the unique spatial characteristics of the graph, comparing the relative locations of data points.

We have hypothesized that users apply the strategies as a function of the task. Certain tasks appear to lend themselves better to one strategy than another. Answering a comparison question like "Which is greater, variable A or B?" would probably be answered rapidly and with high accuracy by comparing the spatial locations of A and B. In contrast, a user answering the question "What is the difference between variables A and B?" about a line graph might be able to apply the perceptual strategy, but would be able to determine the answer more easily and accurately with the arithmetic strategy. In addition, we propose that users vary their strategy according to the characteristics of the graph. For example, if a user were faced with a graph that had inadequate numerical labels on the axes, he or she would be forced to use the perceptual strategy to the greatest extent possible.

We have run a series of experiments to test our hypotheses about graphic processing strategies. The response time data from these experiments are consistent with a model that suggests that users tend to apply the arithmetic strategy, but will shift to the perceptual strategy under certain conditions. In the basic experiment, subjects used three types of graphs -- a scatter plot, a line graph, and a stacked bar graph. They were asked eight types of questions about each graph type: (1) identification -- what is the value of variable A? (2) comparison -- which is greater, A or B?, (3) addition of two numbers -- A+B, (4) subtraction -- A-B, (5) division -- A/B, (6) mean -- (A+B+C+D+E)/5, (7) addition and division by 5 -- (A+B)/5, and (8) addition of three numbers A+B+C. Subjects were instructed to be as fast and accurate as possible. We predicted that the subjects' time to answer the questions using a graph would be a function of the number of processing steps required by a given strategy. Accordingly, with the arithmetic strategy, determining the mean should take longer than adding three numbers, which should take longer than adding two numbers.

We began by fitting the data to a model based on the assumption that subjects used an arithmetic strategy for all questions with all graphs. Figure 1A shows the fit of that model to the response time data. The response time generally increases as the number of processing steps increases, so the model accounts for some of the variance, 61%, but many of the data points fall far from the regression line. This model is poorest at predicting performance on two trials with the stacked bar graph -- the mean and the addition of two numbers -- and for the comparison trials with all three types of graphs; subjects responded on the comparison trials and the the mean trial more quickly than predicted.

As discussed above, a comparison appears to be a likely task for subjects to use a perceptual strategy. In addition, the stacked bar graph intrinsically lends itself to adding the five variables by a perceptual strategy: The total height of the stack represents the cumulative value of the five variables. Accordingly, for model 2, we assumed that subjects used a perceptual strategy to determine the cumulative value of the stacked bar graph (then looked up the value and divided by 5 arithmetically), and used only the perceptual strategy to make all comparisons. Figure 1B shows how a version of that model fits the data. This model captures a substantially greater amount of the variance, 91%, than did Model 1. In this version of the model, the regression function slope suggests that each processing step required about 1 second to complete, except for steps requiring subtraction or division (which the model assumes took 1.5 and 2 seconds, respectively).

The fit of the mixed arithmetic-perceptual model to the data, together with subjects' verbal protocols when answering questions using graphs, support our hypotheses: (1) that people use both arithmetic and perceptual strategies with graphics, (2) that for many typical questions, the bias appears to be for the arithmetic strategy (perhaps because of the greater accuracy with that strategy), and (3) subjects switch strategies as a function of the characteristics of the question and graph.

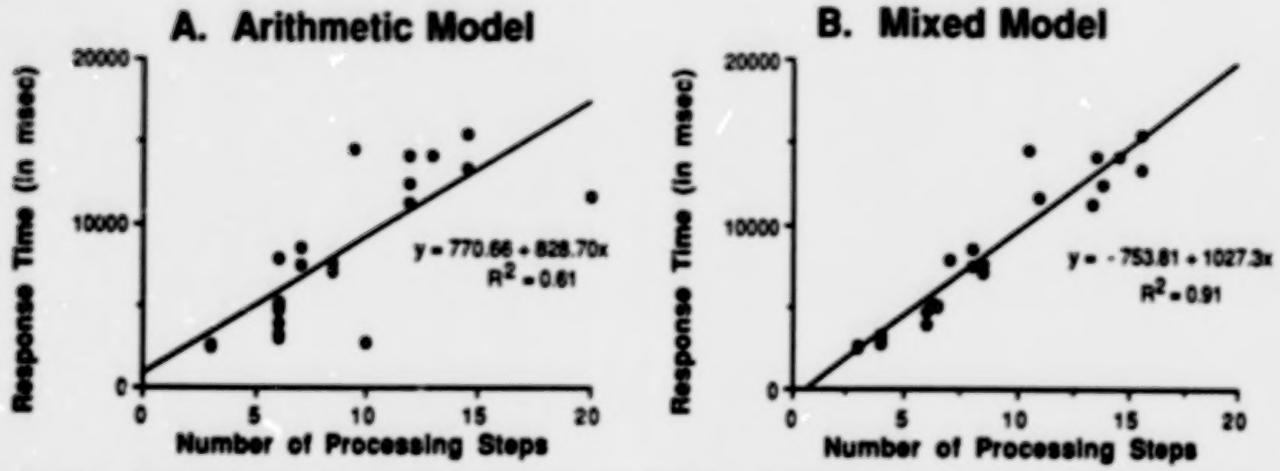


Figure 1. Response times for answering eight types of questions using three types of graphs as a function of the number of processing steps. A. Arithmetic strategy. B. Mixed arithmetic-perceptual strategy.

A THEORY OF GRAPHIC COMPREHENSION

The focus of the rest of this paper is on an overall theory of graphic comprehension designed to help in the development of graphical displays. The theory covers the entire process of graphic comprehension, from the motivation to look at a graph, to the use of the graph, to remembering the graph.

In general, when I look at a graph, I have a particular purpose in mind -- I am usually trying to answer a specific question. Thus, stage 1 in graphic comprehension would consist of either forming a representation of the question to be answered (if the question came from an external information source, for example, a textual question), recalling the question (if the question had to be remembered), or producing the question by inference or generalization. The final cognitive representation of the question would probably be much the same, regardless of whether I read it, remembered it, or generated it. The likely representational format for the question would be a semantic network (e.g., [1], [8]). Determining the answer to the question would function as the goal of my graphic comprehension.

At the start of the second stage in graphic comprehension, I would look at the graph. On looking at the graph, I would encode the primary global features -- the presence or absence of the axes and the type of graph. These would be encoded in a format that would permit reproduction of certain lower level features such as the orientation of both the elements that make up the graph type and the axes. For example, subjects in our representation experiments, generally recalled the horizontal orientation of the bars in a column graph, despite (or, perhaps, because of) their difference from the more typical vertical bar graphs. Interestingly, features that one might expect to be important to a graph user, such as the number of data points, appear not to be encoded as part of this global encoding stage. One hypothesis of this model is that features represented during the global encoding stage receive the bulk of the representational strength. That is to say, they will be the best remembered.

The third stage in graphic comprehension is to use the goal and the global features of the graph to select a processing strategy. If my goal were to compare the value of variables or (possibly) to compare a trend, I would select a perceptual strategy. If my goal were to determine the sum of four variables, numbered axes were present, and the graph type supported it (e.g., a line graph or a bar graph), then I would select the arithmetic strategy.

During the next stage, I would implement the processing steps called for in the strategy determined in the third stage. For example, adding variables A and B from a line graph would involve the following processing steps: 1. Locate the name of variable A on the X axis; 2. Locate variable A in the x-y coordinate space of the body of the graph; 3. Locate the value of variable A on the Y axis and store in working memory; 4. Locate the name of variable B on the X axis; 5. Locate variable B in the x-y coordinate space of the body of the graph; 6. Locate the value of variable B on the Y axis and store in working memory; 7. Add the value of variable A to the value of variable B to produce the value "sum".

Because the semantic and quantitative information (i.e., the variable names and values, respectively) are processed to some extent during this phase, some of that information will be represented, but, as our recall data suggest, not strongly. As a final stage in graphic comprehension, I would examine the result from processing step 7, the "sum" to determine if it plausibly met the goal set in comprehension stage 1. If the response was a plausible fit with the goal, I would incorporate the answer into the semantic network that represented that goal.

This theory directs both future research in graphics and the design of graphical computer interfaces. For example, future research will be needed to determine specific processing models for different questions using the perceptual strategy. In addition, predictions about the memory for quantitative and semantic information in a graph need to be tested. Finally, many of the design

principles derived from the theory are concerned with the complex relations between the task (or goal), the characteristics of the graphical display, and the processing strategies. For example, if a subject is likely to use the arithmetic strategy (e.g., with an addition or subtraction question), the graph type and axes should easily support determining values of specific variables. Accordingly, the axes should be numbered with sufficient numerical resolution. The graph type should allow the user to read a variable's value directly from the axis and should not require multiple computations to determine a variable's value (as a stacked bar graph does). One of our long-term goals is to produce a model of graphic comprehension that is sufficiently elaborate to allow us to build tools to aid in the design of graphical interfaces.

ACKNOWLEDGMENTS

The authors wish to thank Sarah Breedin and Tim McKay for helpful comments on this paper and, along with Kritina Holden and Susan Adam, for helpful discussions about graphics.

REFERENCES

1. Anderson, J. R. THE ARCHITECTURE OF COGNITION, Harvard University Press, Cambridge, MA, 1983.

2. Bertin, J. SEMIOLOGY OF GRAPHICS (translated by W. J. Berg). University of Wisconsin Press, Madison, WI, 1983.

3. Card, S. K., Moran, T. P., & Newell, A. THE PSYCHOLOGY OF HUMAN-COMPUTER INTERACTION, L. Erlbaum Associates, Hillsdale, NJ, 1983.

4. Cleveland, W. S., & McGill, R. "Graphical perception: Theory, experimentation, and application to the development of graphical methods", JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION, 79, 1984, 531-554.

5. Cleveland, W. S., & McGill, R. "Graphical perception and graphical methods for analyzing and presenting scientific data", SCIENCE, 229, 1985, 828-833.

6. Gillan, D. J., Lewis, R., & Rudisill, M. "Models of user interaction with graphical interfaces: I. Statistical graphs", PROCEEDINGS OF CHI, 1989, ACM, New York, 375-380.

7. Hutchins, E. L., Hollan, J. D., & Norman, D. A. "Direct manipulation interfaces", USER-SYSTEM CENTERED DESIGN, L. Erlbaum Associates, Hillsdale, NJ, 1986, 87-124.

8. Kintsch, W., & van Dijk, T. A. "Toward a model of text comprehension and production", PSYCHOLOGICAL REVIEW, 1978, 85, 363-394.

9. Shepherd, R. N., Romney, A. K., & Nerlove, S. B. MULTIDIMENSIONAL SCALING: THEORY AND APPLICATION IN THE BEHAVIORAL SCIENCES, Seminar Press, New York, 1972.

10. Tufte, E. R. THE VISUAL DISPLAY OF QUANTITATIVE INFORMATION. Graphics Press, Cheshire, CT, 1983.

THE SPACE STATION USER INTERFACE LANGUAGE: AN OBJECT-ORIENTED LANGUAGE FOR CONTROLLING COMPLEX SYSTEMS

Dr. R. Davis
University of Colorado

(Paper not provided by publication date.)

LEONARDO AND THE DESIGN OF LARGE DISTRIBUTED SOFTWARE SYSTEMS

K. Fairchild and G. Meredith
MCC

(Paper not provided by publication date.)

EVALUATION OF OFF-ROAD TERRAIN WITH STATIC STEREO AND MONOSCOPIC DISPLAYS

John P. Yorchak
Craig S. Hartley

Martin Marietta Astronautics Group
P.O. Box 179, MS H4371
Denver, Colorado 80201
(303) 971-5748

ABSTRACT

The National Aeronautics and Space Administration is currently funding research into the design of a Mars "rover" vehicle. This unmanned rover will be used to explore a number of scientific and geologic sites on the Martian surface. Since the rover can not be driven from Earth in real-time, due to lengthy communication time delays, a locomotion strategy that optimizes vehicle range and minimizes potential risk must be developed. In order to assess the degree of on-board artificial intelligence (AI) required for a rover to carry out its' mission, we have conducted an experiment to define a "no AI" baseline. In our experiment 24 subjects, divided into stereo and monoscopic groups, were shown video snapshots of four terrain scenes. The subjects' task was to choose a suitable path for the vehicle through each of the four scenes. Paths were scored based on distance travelled and hazard avoidance. Study results are presented with respect to; (1) risk versus range, (2) stereo versus monocular video, (3) vehicle camera height, and (4) camera field-of-view.

INTRODUCTION

The success of the Viking landers on Mars is well documented, but these missions served to raise as many new questions about the Martian surface as they answered. To attempt to answer some of these questions, a broader, more comprehensive exploration of the Martian surface has been proposed. A mission such as this will collect samples from a number of scientific and geologic sites using a "roving" vehicle. This vehicle, capable of some autonomous operation, will be required to navigate the rough and unpredictable terrain of Mars.

A Mars rover can not be driven in real-time from Earth due to communication time delays ranging anywhere from nine to forty minutes, the limited availability of windows for data transmissions, and the limitations of data rates for distances such as this. Therefore, a locomotion strategy that optimizes vehicle range and minimizes potential risk to the vehicle must be developed.

When we speak of the range of the vehicle we must consider several complex issues. If the vehicle requires repeated commands from Earth to travel between sampling sites, it's total range on the surface will be reduced. If on the other hand, the vehicle is capable of autonomous moves, due to onboard artificial intelligence (AI) and a superior ranging and vision system, then the vehicle's range and it's ability to gather more samples, is maximized. Unfortunately, the cost of this autonomous capability is extremely high.

In order to begin to assess how much AI the rover needs onboard to carry out its' mission, we have designed a study to define a "no AI" baseline. This study will evaluate a Mars rover scenario where the vehicle possesses no onboard intelligence, and therefore, would be teleoperated from Earth. With this scenario, the only visual information available to the operator concerning the Martian surface would be a series of video snapshots from the rover's cameras. The key question we have addressed with this experiment is, using only video snapshots, is it possible for an operator to plan a safe path for the vehicle through hazardous terrain, while at the same time, maximizing each discrete vehicle move?

If we look at two operational extremes this question should become a little more clear. On one hand, as an operator of the vehicle I am fairly certain that I can steer clear of hazardous terrain if I move the vehicle one meter at a time. However, since I will only be able to make one move approximately every 40 minutes, the vehicles total range per day would only be 24 meters. If, on the other hand, I take some risks, I might be able to move the vehicle 20 meters at a time. With this scenario I would be able to move 480 meters per day, but the element of risk may be unacceptable.

Using video snapshots of terrain around the main plant at Martin Marietta Astronautics in Denver, Colorado, we have created four scenes that contain differing degrees of perceived hazards and actual hazards. Using these four scenes as stimulus displays, subjects were asked to draw the safest and most direct path through the

scene that the vehicle could take. Stereo and monocular snapshots were presented to equal numbers of subjects.

METHOD

Subjects

Two groups of twelve subjects completed the experiment. The subjects were all employees of the Martin Marietta Astronautics Group who volunteered for the experiment. The subjects' positions ranged from upper management to lab technician.

Apparatus

Control Console

This experiment was conducted using one of the Space Operations Simulator (SOS) Laboratory Advanced Technology Control Consoles. These consoles were designed as testbeds for such advanced human/system interface technologies as touchscreens, programmable display pushbuttons, speech recognition, speech synthesis, expert system workaids, hypermedia systems, integration of computer graphics and real-time video imagery, and stereoscopic video displays. As testbeds, these consoles are routinely used to evaluate a wide range of technological innovations in human interface design in the context of real-time task simulations conducted by the SOS Lab.

Structure. The control console configuration used for this experiment is shown in Figure 1 below. A custom design metal structure encloses all of the console components. The structure is optimized to match the anthropometric requirements for 50th percentile male operators. However, the console design can accommodate users ranging from 5th percentile females through 95th percentile males through the provision of a special adjustable chair, adjustable handcontrollers, arm rests, and foot pedals. All controls and displays are positioned and angled to ensure easy access and clear direct visibility from a reference eyepoint at the operator's seated position. Only the lower center monitor was used for this experiment.

For the experiment described here, subjects were asked to sit at a comfortable viewing distance (typically 16 to 18 inches) from the primary, lower-center monitor. They adjusted their chair height and position so that their eyes were normal (at 90 degrees) to the lower center monitor. Handcontrollers were installed but were not used as part of the experiment.

Lower Center (Stereoscopic Display) Monitor. The lower center control console monitor was used as the display for this experiment. It is a special 16-inch Tektronix high-resolution, fast refresh (120 Hz), fast phosphor, color CRT designed for real-time stereoscopic or monoscopic video display. This monitor is optionally equipped with a touchscreen when the stereo system is not in use, but normal control

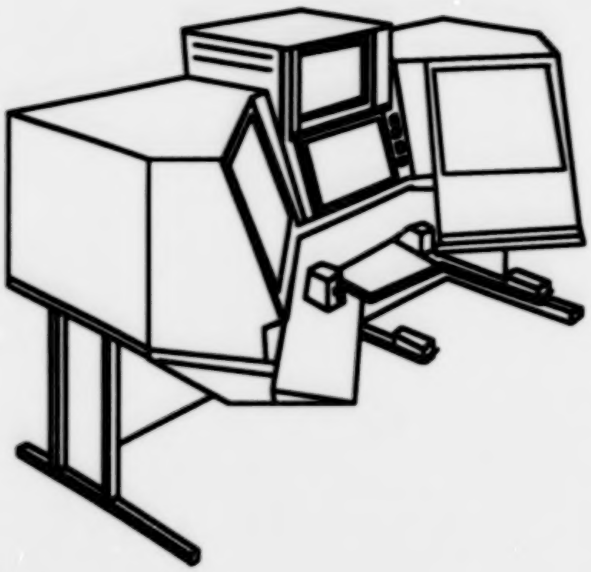


Figure 1. Control console configuration.

of on-screen functions (such as graphics reticle overlay selection) is provided either by programmable LED or EL display pushbuttons located to the right of the monitor or by voice command. The monitor display is driven by Parallax 1280-V-8VN-TS high resolution (1280 x 1024) videographic processor boards mounted in a Sun 3/260 computer.

The stereoscopic video display system uses a time serial presentation of left and right eye views in synch with shifts in polarity of a Tektronix Liquid Crystal Display (LCD) Shutter overlay. The shutter is circularly polarized to allow consistent stereo viewing even when the viewer's head is tilted. A small, horizontal, and nearly invisible seam divides the middle of the screen because it consists of two pieces to facilitate high-speed (120 Hz) switching of the LCD polarity. The shutter is switched by TTL command line from the host computer.

Video images and synchronized TTL switching signals for the LCD shutter are generated by a Parallax Graphics 1280 -V-8VN-TS video graphic board set mounted in a VME bus 21-slot expansion chassis connected to the Sun 3/260 computer. The Parallax video graphics boards provide for input of two NTSC camera views (from the left and right cameras). As the video images are received, the boards digitize each NTSC image into a 640X480 color bit-mapped image (in near real-time) and stores it in a separate RAM buffer. To display stereo, the system alternately sends the bit-mapped image for left and right eye views. At the same time it sends a synchronized TTL command signal to the LCD shutter to switch the polarity as appropriate for each eye view. The operator wears special glasses having each eye lens separately cross polarized so that he/she only sees the left eye view when the digitized image from the left eye is displayed and vice versa. Special software allows shifting of the left and right eye image pixel arrays so that all

undesirable vertical and horizontal disparity could be eliminated. This capability allowed us to compensate for slight mechanical misalignments inherent in the camera mounts. The system is capable of maintaining a 15 frame/sec./eye (30 frames/sec. for both eyes) rate with a 640X480 pixel full color image refreshed at 120 Hz. Resolution can be traded for frame rate so it is possible to display a 512 X 240 pixel image at 30 frames/sec./per eye.

The stereo display condition used as an independent variable for this experiment consisted of a static (not moving) 640 X 480 color presentation at 15 frames/sec./eye with a 120Hz refresh rate. The image was previously digitized and stored on hard disk. The image was clear, free of any flicker, and not discernable from live video of the same scene. The stereo effect was excellent and very realistic, and represented the state-of-the-art in stereoscopic display technology.

The monoscopic condition was created by displaying a single (right eye) view from the stereo pair. This image was also flicker free and was identical in appearance to the stereo condition. The only difference in the display was the slight horizontal shift in picture location associated with a single eye view in a stereo pair.

Cameras

The video input devices used for this study consisted of a matched pair of Panasonic CCD high resolution color cameras equipped with Panasonic motor drive auto iris lenses. These systems provided a 57 degree field-of-view when fully zoomed back to their widest angle (the setting used in this experiment).

The cameras were mounted on a custom-designed, computer-controller, mechanism that provided stepper motor control to 0.1 degree in pan, tilt, and camera convergence. The interocular separation between cameras was manually adjustable. Prior to the start of this experiment we carried out a number of pilot studies to establish an optimized stereoscopic picture using this system. Based on that research, literature reviews, consultations with stereo system experts, and the sizes of the system components, we set the interocular distance at 6".

Stereo Camera Setup

Discussions with Dr. Ed Spain at the Naval Oceans System Center (NOSC) and Dr. Robert Cole at the University of Hawaii regarding an optimal stereo camera setup for a remotely controlled vehicle, indicate that an interocular distance of 2.5 - 3", and cameras as close to parallel as possible, is the best setup for distances greater than 100 ft. When the area of interest is 50 - 75 ft. away, a separation of 7.5 - 10" is recommended when the cameras are converged at a point just in front of the "target". For this study, the ALV's stereo camera pair were separated approximately 6", and converged at a point approximately 50 ft. in front of the vehicle. This configuration was used for all four scenes.

As seen in Figure 2, the stereo pair mechanism was mounted on a platform 3 feet above the roof of the ALV, making the camera positions it approximately 15 ft. above the ground. In order to create the widest field of view possible, the cameras were zoomed out all the way (creating a 57 degree horizontal field of view), and the camera platform was tilted down slightly. The bottom of the field-of-view in the console monitor was 24 ft. in front of the vehicle.

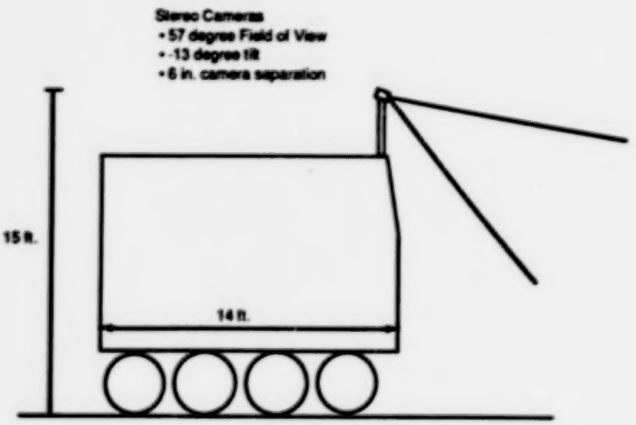


Figure 2. Location and set-up of ALV's stereo camera pair.

Terrain Snapshots

A series of pilot studies were conducted after the stereo camera set up was optimized and prior to the actual experiment to select four test sites that met the following conditions:

Scene 1: actual hazard present in the scene even though it appears relatively safe to the operator,

Scene 2: no actual hazard present in the scene even though the scene appears hazardous,

Scene 3: no actual hazard present in the scene and no appearance of any, and

Scene 4: actual hazard present in the scene and the scene appears hazardous.

Once the sites were selected, video snapshots were made of each by poitioning the vehicle, digitizing the left eye and right eye camera images, and saving those images to disk for later redisplay. All four scenes were digitized on the same day and within a one hour time period so that lighting and weather conditions were the same for each image.

Experimental Procedure

As the subjects arrived at the lab they were greeted by the experimenter and seated at the console so that their eyes were at a standard height and position, with respect to the display (approximately 18" from the screen). They were given the stereo glasses to wear (if they were in the stereo group) and then given a few

minutes to view a sample scene in order to get acclimated to the stereo.

As the subjects viewed this scene the experimenter read them the following briefing:

> *This experiment is being conducted to investigate human remote-driving of the Mars Rover vehicle using stereo (video) snapshots. Since a Mars Rover doesn't currently exist, we are conducting these studies with the functionally similar Autonomous Land Vehicle (ALV). We are attempting to determine the average distance the driver of the Mars Rover would move the vehicle, given only stereo (video) snapshots of the scene. We are also interested in the length, and driver confidence, of these moves given the fact that the scenes often contain unknown hazards.*
>
> *Your task will be to view 4 different stereo (video) snapshots. After studying the scene for a few moments, you will be asked to draw a path for the vehicle to travel on a line drawing of the scene. This path should be the safest path possible that would take the vehicle from its' current position through the terrain displayed in the snapshot. After drawing the path, place an "X" on the point where you would like the vehicle to make its' first stop to send you a new display. The location of this "X" should represent the point on the path that, given only the information provided by the snapshot, you believe the vehicle could reach without running into any hazards in the terrain.*
>
> *Are there any questions so far?*
>
> *Before we start the actual experiment we will do a practice trial on the scene displayed in front of you.*
>
> *Take a few moments to plan a path through the scene. When you have a path in mind say "OK".*
>
> (After the subject says "OK", give him/her a line drawing of the scene to mark on.)
>
> *Mark your path on this drawing, and then mark your "X" where you would like the vehicle to stop and send you a new display.*
>
> *If you don't have any other questions, we're ready to begin the experiment.*

Since the subjects had the actual video display in front of them as they used the hardcopy drawing of the scene, they had no trouble drawing their path and positioning their "X" on the picture. We deliberately chose to use a "poor" quality picture because we did not want to add to the subjects' knowledge of the terrain in any way.

In order to be sure that the subjects fully understood their task, the experimenter and the subject worked through the procedure described above using a sample scene displayed on the monitor. This practice scene consisted of a view down a gravel road with hills, telephone poles, and a few bushes and trees in adjacent fields.

An experimental trial consisted of the experimenter sending the appropriate scene to the subjects' monitor from a separate experimenter's workstation, and at the same time starting a timer. After the subject had taken as much time as he/she needed to study the scene and plan a path through it, they told the experimenter that they had a path in mind. The experimenter then stopped the timer and gave the subject the hardcopy picture to draw on. After they returned the marked-up drawing to the experimenter, the next scene was displayed on the subject's monitor. To guard against the possibility of an order effect confounding the data, the presentation order of the scenes was counterbalanced across subjects.

After completing all four experimental trials, the subject completed a brief questionnaire asking about his or her sex, age, position at Martin Marietta, eyesight, experience with stereo vision systems, and experience with remotely operated vehicles (including remotely controlled hobby cars).

After completing the questionnaire each subject was thoroughly debiefed. This debriefing consisted of redisplaying the scenes and soliciting comments about the subjects rationale for their path and placement of their "X". Subjects in the monoscopic group were shown the scenes again, this time in stereo. Many of them remarked that stereo made the scenes look different, but not different enough to change their paths.

Independent Variables

The experimental design used in this study was a mixed factor 2X2X2 repeated measures analysis of variance, where the display viewed, stereo or monocular, was the between subjects factor, and the two within subjects factors were presence of hazard and appearance of hazard. Table I illustrates this experimental design.

Other data, such as sex, age, eyesight, and related experience, gathered from the questionnaire were analyzed with a regression model.

Dependent Variables

Data for three dependent variables were collected and analyzed. These were; (1) the amount of time the subject took to study the scene from the time it was first displayed to when he/she said "OK", (2) the actual distance travelled to the "X", and (3) a driver "efficiency" score derived from a qualitative assessment of the path's direction, distance to the "X", and degree of hazard avoidance.

Data Analysis

Each of the dependent measures was analyzed with a 2X2X2 repeated measures analysis of variance model. The statistical package used to analyze the data was Systat version 3.1 running on a Macintosh II.

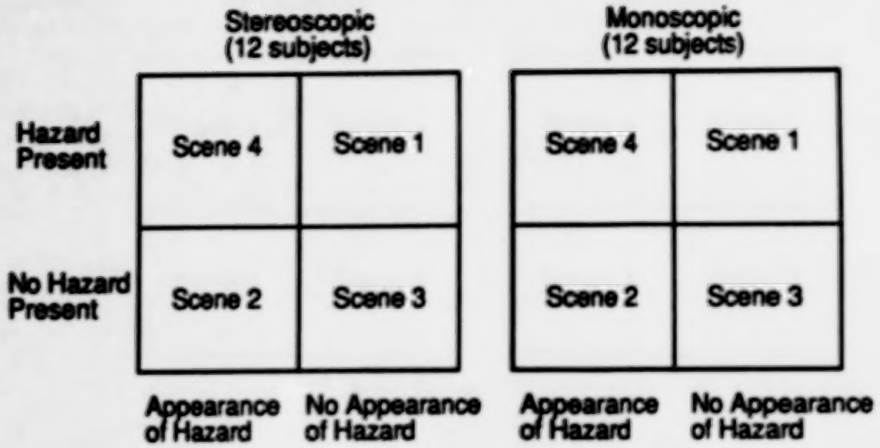


Table I. Experimental design.

RESULTS

Decision Times

The amount of time subjects took to decide on a path was collected by having the experimenter start a timer once the scene was displayed on the subject's monitor, and then stop it when the subject indicated to the experimenter they had a path.

The results of the analysis of variance indicate that display type did not significantly effect decision times, $F(1,22) = 0.803$, $p<.380$. However, the main effect of appearance of hazards in the scene did, $F(1,22) = 20.259$, $p<.000$. There was also a significant interaction between this main effect and display type, $F(1,22) = 4.383$, $p<.048$. The other main effect of hazard in the scene was not significant, $F(1,22) = 2.595$, $p<.121$. Actual decision times are shown in Table II.

	Appearance of Hazard	No Appearance of Hazard
Hazard Present	Scene 4 (54.2)	Scene 1 (38.5)
No Hazard Present	Scene 2 (51.0)	Scene 3 (29.0)

(Note: Times in parentheses are in seconds.)

Table II. Decision times collapsed across display type.

Distance Traveled to the "X"

The analysis of the distances subjects traveled to the "X" indicate that display type made no difference in their decisions, $F(1,22) = 0.006$, $p<.941$. However, as with decision times, a main effect was observed for the appearance of hazards in the scene, $F(1,22) = 37.815$, $p<.000$. In addition, a main effect for hazards in the scene was also significant, $F(1,22) = 15.273$, $p<.001$. Average distances collapsed across display type are shown in Table III.

	Appearance of Hazard	No Appearance of Hazard
Hazard Present	Scene 4 (22.3)	Scene 1 (32.5)
No Hazard Present	Scene 2 (13.0)	Scene 3 (97.3)

(NOTE: Distances in parentheses are in feet.)

Table III. Distances collapsed across display type.

Efficiency Score

The efficiency score was derived by drawing a grid over a picture of each scene. Each cell of this grid was given a score between 0 and 10 for each of three factors. Factor 1 was the distance driven through the scene, factor 2 was the direction, and factor 3 was the degree of hazard avoidance. Therefore, if a subject placed his/her "X" in the optimal position on a scene, then that cell of the grid was worth 30 points. The analysis conducted on the driver efficiency scores revealed that display type did not significantly effect these scores, $F(1,22) = 0.247$, $p<.624$. A main effect was observed, however, for the appearance of hazards in the scene, $F(1,22) = 63.716$, $p<.000$. The main effect for actual hazards was not significant, but the interaction between appearance and actual hazards was, $F(1,22) = 7.949$, $p<.010$. These scores have been normalized so that a score of 30 represents the best possible score available on <u>each</u> of the four scenes. Therefore, as can be seen in Table IV, the appearance of a hazard in the scene apparently caused subjects to choose paths that were less than optimal.

	Appearence of Hazard	No Appearence of Hazard
Hazard Present	Scene 4 (19.46)	Scene 1 (23.42)
No Hazard Present	Scene 2 (18.21)	Scene 3 (27.67)

Note: The values in parentheses may range from 30 (Best) to 0 (Worst).

Table IV. Efficiency scores collapsed across display type.

Terrain Scenes

A detailed analysis of each scene is required if we are to fully understand the significance of the anovas.

Scene 1

The terrain in this scene is essentially hazard-free in the lower two-thirds of the scene, but the upper third contains an unknown area that most subjects perceived as a potential hazard. Since the majority of the subjects perceived a hazardous area in the scene, it may not have been the best scene for the "No Appearance/Hazard Present" scene. However, five out of our twenty-four subjects (21%) apparently did not see the hazard because they put their "X" in it. Of the 19 who stayed short of the hazard, their average movement was 27 ft., or approximately 75% of the total distance they could have moved. Figure 3 shows the actual distances that subjects moved in this scene.

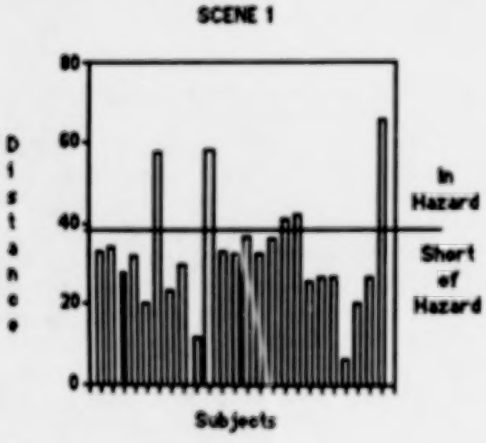


Figure 3. Distance moved by each subject in Scene 1.

Scene 2

Scene 2 was composed to be a hazardous-looking scene that actually contained no hazards. Since there were no actual hazards present in this terrain it is impossible to say whether or not subjects made "risky" decisions regarding their path and distance. However, it is obvious from Figure 4 that most subjects made very conservative decisions regarding this scene. Their difficulty arose from an area directly in front of the vehicle that appeared to be a ditch or a wash. Actually, this area is perfectly safe, with the deepest depression being less than 3" deep. As can be seen in Figure 4, all subjects except one steered around this "problem" and kept their first move quite short. Figure 4 shows the actual distance moved by each subject. The maximum distance possible was 80 feet.

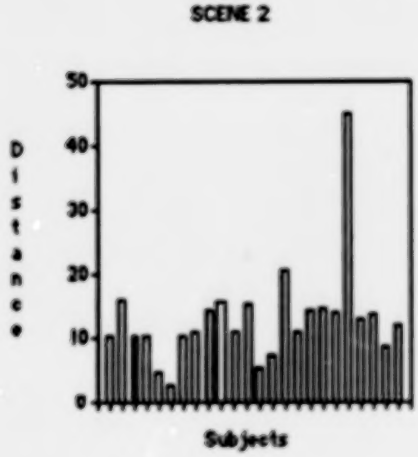


Figure 4. Distance moved by each subject in Scene 2.

Scene 3

Scene 3 was composed over a flat, hazard-free section of the test area, so that it would appear hazard-free to the subjects. It appears from the data that this was the perception of most of the subjects. However, not all subjects perceived the scene as totally hazard-free. As Figure 5 indicates, 13 out of 24 subjects (54%) made conservative judgements regarding this terrain. Figure 5 shows the actual distance moved by each subject. Note that five subjects went as far as they could through the scene.

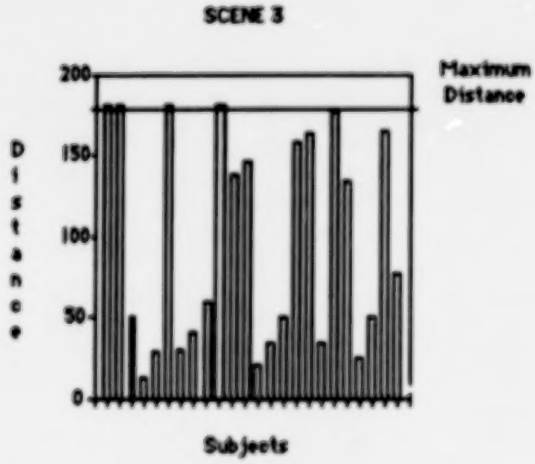


Figure 5. Distance moved by each subject in Scene 3.

Scene 4

Scene 4 was composed to appear hazardous to the subjects while actually being quite hazardous to a vehicle. The hazards present in the scene cut across it at a 30 degree angle to the horizontal. These hazards consisted of a number of channels and washes that would cause a great deal of difficulty for a vehicle. Most subjects perceived the hazards in the upper half of the scene, but they also mis-classified areas directly in front of the vehicle as hazardous when in fact they weren't. Figure 6 shows the actual distance moved by each subject in this scene.

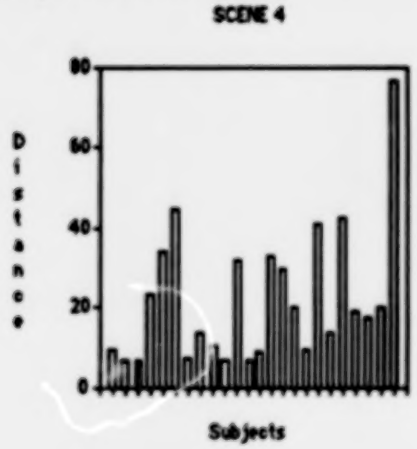


Figure 6. Distance moved by each subject in Scene 4.

DISCUSSION

Risk and Range

Many interpretations of the data presented in this study are possible but we will concentrate on two basic issues---risk and range. With the exception of Scene 1, where five subjects put their "X's" in the hazard, subjects were generally very cautious with their path and distance decisions. It seems that they erred more often in a cautious direction than a careless one. This result is consistent with a finding reported by Spain (1987). In this study he reported that experienced teleoperators of off-road vehicles tended to make fewer errors, but also drove slower. Spain concluded that the effect of training may be learning caution.

Erring on a cautious side would seem to be desirable for operational scenarios involving the Mars rover. However, overly cautious operators significantly decrease vehicle range. In the case where a Mars rover has a limited life on the surface of the planet, it is critical to mission success to maximize each movement. As stated previously, and seen in the scenes that were evaluated, even when hazards were not present in the scene operators often avoided anything that even looked hazardous. This result indicates a need for on-board sensing and ranging of the ground surface. A locomotion strategy that appears reasonable given our findings has been proposed by Wilcox, Salo, Cooper, and Killon (1986). In their study, the operator of a computer-aided remote driving system was asked to identify path points from viewing a 3-dimensional video display of the area in front of the vehicle. Given the path points the operator selected, the computer system commanded the vehicle to execute the path. With this approach the operator is removed from the details of vehicle control, while the computer is removed from the tasks of scene analysis, global path planning, and obstacle avoidance. Testing with this system has indicated that operation over path segments of up to 40 meters has been demonstrated successfully.

Stereo vs. Monocular

It is somewhat surprising that stereo did not aid subjects in the interpretation of terrain scenes. In similar studies, stereo has generally outperformed monocular displays (Spain, 1987). One important difference between this study and those reported by Spain, however, is that our subjects only had single, static frames of stereo or monocular video, rather than real-time dynamic displays. This would seem to indicate that the advantage of stereo lies in the depth cues provided by the relative motion between objects in the foreground and background. When motion is not presented, even though the stereo snapshots "look" better than comparable monocular snapshots, the lack of motion cues on the stereo displays makes them comparable to monocular views.

Discussions with Dr. Robert Cole at the University of Hawaii have revealed another interpretation of our failure to find an advantage for stereo. The height, field of view, and downward tilt of the stereo cameras are critical to the overall stereo effect created. With our stereo pair mounted on top of the ALV approximately 15 ft. above the ground and tilted down 13 degrees, the overall scene was "flattened" to such an extent that the stereo effect was essentially eliminated. This finding is important for future work involving the ALV's stereo system, whereby, in order to optimize the stereo effect provided by the ALV's cameras, it may be necessary to lower the camera platform considerably.

Spain (1987) also concluded that when experienced and inexperienced drivers of a teleoperated vehicle drove with stereo and monocular video, no significant difference was found for stereo, "... one must remember that past research has shown that the advantages stereoscopic imagery provides are most pronounced in unfamiliar, visually cluttered and in visually degraded scenes. Stereoscopic imagery is also useful in judging the relative distances and orientations of objects and terrain surface features - all of which might prove invaluable to an operator in "reading" terrain before attempting to traverse it."

Camera Height

When the stereo pair is too high off the ground, it is impossible to see the horizon and the ground directly in front of the vehicle simultaneously. If the horizon is centered on the display, then a considerable distance in front of the vehicle is not in view. Conversely, if the

ground in front of the vehicle is important, the cameras must be tilted down so dramatically that the horizon is not in view and the terrain that is in view becomes "flattened".

These results can easily be applied to Mars rover. Given our finding that stereo is adversely effected when the camera pair is too high off the ground, future studies should be done in order to define the optimum height of the rovers' stereo cameras.

Camera Field of View

Field of view is very important to operators of remote vehicles. As mentioned previously, our cameras were set at the maximum horizontal field of view possible, 57 degrees. This field of view seemed adequate but the height and downward tilt of the ALV's cameras had the effect of causing 22 ft. directly in front of the vehicle to be out of view of the operator. An interaction of camera field of view, height, and tilt determine how much of the terrain an operator is able to see. When a 16 mm lens (approximately 31 degrees horizontal field of view) was compared to a 4 mm lens (approximately 96 degree horizontal field of view) for teleoperation of a small vehicle in an indoor environment, there was a significant reduction in the number of times the vehicle touched obstacles with the 4 mm lens (Silverman, 1982). In a similar study by Horst, Rau, LeCocq, and Silverman (1983) using these same lenses, stereo viewing improved performance (number of obstacles bumped) for both camera lenses.

ACKNOWLEDGMENTS

This research was performed under Martin Marietta Astronautics Group IR&D task, "Planetary Rover Technology."

REFERENCES

Frederick, D.K., "An Autonomous Roving Vehicle For the Planet Mars," Technical Report MP-83, Rensselear Polytechnic Institute, Troy, NY, October 1984.

Horst, R.L., Rau, P.S., LeCocq, A.D., and Silverman, E.B., "Studies of Three-Dimensional Viewing in Teleoperated Systems," Remote Systems Technology Supplement to the Transactions of the American Nuclear Society, Detroit, Michigan, Summer Meeting, 1983.

Silverman, E.B., "Robotic Technology Experiments for Nuclear Power Plant Inspection and Maintenance," Transactions of the American Nuclear Society, Volume 43, 1982.

Spain, E.H., "Assessments of Maneuverability with the Teleoperated Vehicle (TOV)," Presented at the Fourteenth Annual Symposium of the Association for Unmanned Vehicle Systems, Washington, D.C., July 19-21, 1987.

Wilcox, B.H., Salo, R., Cooper, B., and Killon, R., "Computer-Aidad Remote Driving," Thirteenth Annual Meeting of the Association for Unmanned Vehicle Systems, Boston, Massachusetts, July 21-23, 1986,.

OCEAN FEATURE RECOGNITION USING GENETIC ALGORITHMS WITH FUZZY FITNESS FUNCTIONS (GA/F^3)*

by

C.A. Ankenbrandt[1], B.P. Buckles[1], F.E. Petry[1], & M. Lybanon[2]

[1]Department of Computer Science
Center for Intelligent and Knowledge-based Systems
301 Stanley Thomas Hall, Tulane University
New Orleans, LA 70118, (504) 865-5840

[2]Remote Sensing Branch
Naval Ocean Research and Development Activity
NSTL Station, MS 39529

ABSTRACT

A model for genetic algorithms with semantic nets is derived for which the relationships between concepts is depicted as a semantic net. An organism represents the manner in which objects in a scene are attached to concepts in the net. Predicates between object pairs are continuous valued truth functions in the form of an inverse exponential function ($e^{-|x|}$). I:n relationships are combined via the fuzzy OR (Max [...]). Finally, predicates between pairs of concepts are resolved by taking the average of the combined predicate values of the objects attached to the concept at the tail of the arc representing the predicate in the semantic net. The method is illustrated by applying it to the identification of oceanic features in the North Atlantic.

keywords: genetic algorithms, feature labelling, semantic nets, fitness functions

* This work was supported in part by a grant from Naval Ocean Research and Development Activity, Grant #N00014-89-J-6003

BACKGROUND

Genetic algorithms are a problem solving method requiring domain-specific knowledge that is often heuristic. Candidate solutions are represented as organisms. Organisms are grouped into populations known as generations and are combined in pairs to produce subsequent generations. An individual organism's potential as a solution is determined by a fitness function.

Fitness functions map organisms into real numbers and are used to determine which organisms will be used (and how frequently) to produce offspring for the succeeding generation. Fitness functions often require heuristic information because a precise measure of the suitability of a given organism (i.e., solution) is not always attainable. An example is the recognition (i.e., labeling) of segments in a scene. General characteristics of objects in the scene such as curvature, size, length, and relationship to each other may be known only within broad tolerance levels. That is, there is great variability in the relationships among objects in different scenes.

Semantic nets (SNs) are effective representations of binary relationships between concepts (e.g., objects in a scene). SNs denote concepts via nodes in a directed graph. The arcs are labelled by predicates. We introduce here a representation of an organism whose fitness function evaluation is dependant upon an SN context.

Because relationships (i.e., predicates) relating concepts are not precise, their evaluation is in the form of a truth functional with range [0,1] rather than the traditional {0,1}. That is, we use fuzzy logic [YA75, ZA88, ZI85] to combine heuristically the information concerning a particular organism. Thus, we derive genetic algorithms with fuzzy fitness functions (GA/F^3).

GENETIC ALGORITHMS

Genetic algorithms (GAs) are search procedures modelled after the mechanics of natural selection. They differ from traditional search techniques in several ways. First, GAs have the property of implicit parallelism, where the algorithm is equivalent to a search of the hyperplanes of the search space, without directly testing hyperplane values [HO75, GO88]. Nearly optimal results have been found by examining as few as one point for every 2^{35} points in the search space [GO86]. Second, GAs are randomized algorithms, using operations with nondeterministic results. The results for an operation depend on the value of a random number. Third, GAs operate on many solutions simultaneously, gathering information from all current points to direct the search. This factor mitigates the problems of local maxima and noise.

From a mechanistic view, genetic algorithms are a variation of the generate and test method. In pure generate and test, solutions are generated and sent to an evaluator. The evaluator reports whether the solution posed is optimal. In genetic algorithms, this generate and test process is repeated iteratively over a set of solutions. The evaluator returns information to guide the selection of new solutions for following iterations.

GA terminology is taken from genetics. Each candidate solution examined is termed an organism, traditionally represented as a list. The set of organisms maintained is termed a population, and the population at a given time is termed a generation. Each iteration envolves three steps.

First, each organism in the current generation is evaluated, producing a numerical fitness function result. The criteria for evaluation is domain specific information about the relative merit of that particular organism. Better organisms are assigned higher fitness function values. Second, some organisms are selected to form one or more organisms for the next generation. Specifically, the number of copies of each organism selected is directly proportional to its fitness function. Third, some of those organisms selected are modified via genetic operators. Each genetic operator takes the chosen organism(s), and produces a new organism(s). The most common genetic operators include crossover and mutation. This iterative procedure terminates when the population converges to a solution.

The crossover operator takes two organisms selected and combines partial solutions of each. When organisms are represented with lists, single point crossover can be viewed as combining the left hand side of one organism chosen with the right hand side of the other, and conversely. This creates two offspring. The crossover point, that point where the crossover takes place, is randomly determined.

The mutation operator uses a minimal change strategy. It takes a selected organism, and changes the value at one randomly determined position. This corresponds to a tight local search. The offspring produced is identical to the parent except at the mutation point.

GENETIC ALGORITHM PROBLEM MODEL FOR OCEANIC FEATURE LABELING

Scene recognition is an application for which the GA model we propose is suited. For example, Fig. 1(a) is a segmented image of the North Atlantic for which Fig. 1(b) is the original image. The lines (referred to here as segments, s_1, s_2, ...) represent boundaries between warm and cold regions of sea water. The problem is to classify the segments as Gulf Stream North Wall (NW), Gulf Stream South Wall (SW), cold eddies (CE), warm eddies (WE), continental shelf (CS), and "other" (O).

Relationships which can be expressed as fuzzy truth functions are known to exist within or between classifications. Principal among these are (1) the average width of the Gulf Stream is 50 kilometers, (2) the average diameter of an eddy is 100 kilometers, (3) cold eddies are usually south of the Gulf Stream, and (4) warm eddies are usually north of the Gulf Stream. To these one must add the trivial (yet necessary) relationships such as the south wall is at a lower latitude than the north wall and the known geophysical coordinates of continental shelves.

A scene consisting of classification categories (cat_1, cat_2, ..., cat_n) and relationships expressed as truth functions ($p^{(1)}_{ij}$, $p^{(2)}_{ij}$, ...) between categories can be modelled as a semantic net (or, more precisely, an association list). A generic one is shown in Fig. 2. Segments are



a. Segmented Image

b. Original Infrared Image

Figure 1. Oceanic Features (North Atlantic)

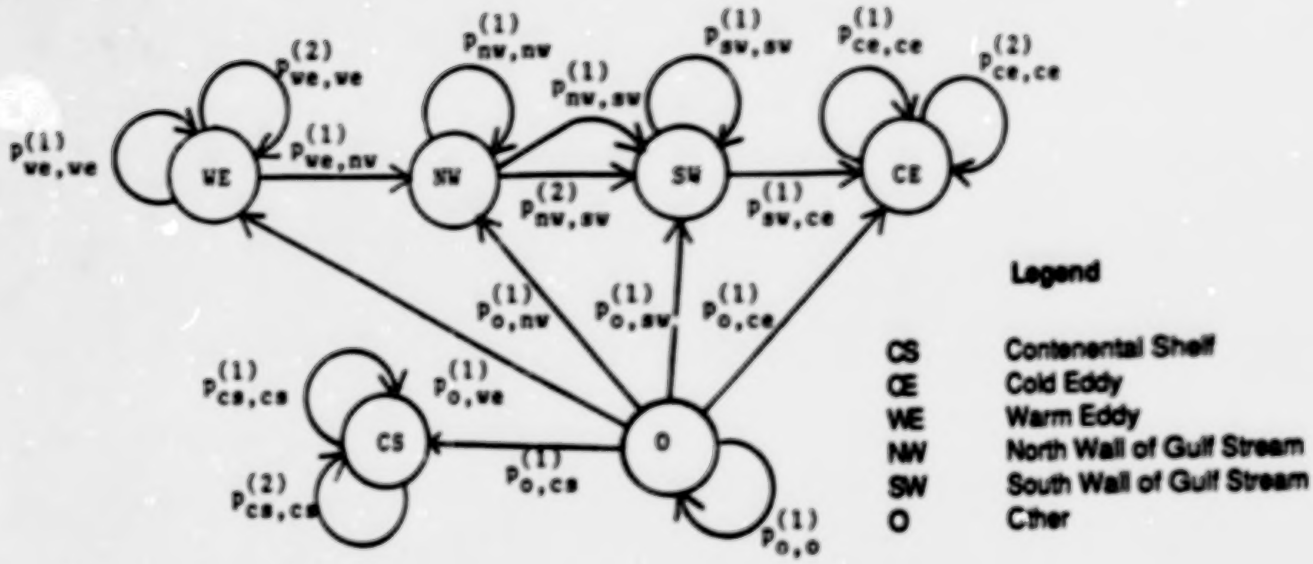


Figure 2. Generic Semantic Net for Oceanic Features

attached to the categories via the INST (instance) relation. An allele (or gene) is a category name. An organism is a list of categories, one allele for each segment. For example, given six segments then (NW, NW, SW, CS, CE, O) and (CE, SW, CE, O, O, CS) are representative organisms. Formally, let an association list be defined as A = <V,P> where V = $(cat_1, cat_2, \ldots, cat_m)$ is a set of categories, and P = $(p_{ij}^{(g)} \mid i,j \leq m, g = 1,2,\ldots, r_{ij})$ is a set of binary predicates. These predicates describe the relationships between categories and the ideal relationship between segments assigned to these categories.
Let an organism for spatial labeling is defined as O = <S, INST>, where S = $(s_1, s_2, \ldots, s_n)$ is a set of segments, and INST: S -> V is a function.

Crossover Operators

There are three applicable crossover operators. These include single point crossover, two point crossover, and varying multiple point crossover [DO87]. Crossover operators require the imposition of a total order on the segments in S. Let $s_i < s_j$ if $i < j$; $s_i = s_j$ if $i = j$; $s_i > s_j$ if $i > j$. Denote by $INST_{Oi}$ the instance mapping for organism O_i.
Single Point Crossover. Given $< s_1, s_2, \ldots, s_n>$, choose a random integer k, $1 \leq k < n$. For parent organisms O1 and O2 create an offspring, O', such that

$$INST_{O'}(s_i) = \begin{cases} INST_{O1}(s_i) & \text{if } i \leq k \\ INST_{O2}(s_i) & \text{if } i > k \end{cases}$$

Two Point Crossover. Let $< s_1, s_2, \ldots, s_n>$ be a circular list. Formally, $succ(s_i) = s_{i+1}$ ($pred(s_{i+1}) = s_i$) if $i < n$ and $succ(s_n) = s_1$ ($pred(s_1) = s_n$). Choose two random integers, k1 and k2. For parent organisms O_1 and O_2 create an offspring, O' such that

$$INST_{O'}(s_i) = \begin{cases} INST_{O1}(s_i) \text{ if } s_i \in (s_{k1}, succ(s_{k1}), \ldots, pred(s_{k2})) \\ INST_{O2}(s_i) \text{ otherwise} \end{cases}$$

Varying Multiple Point Crossover. For parent organisms O_1 and O_2, create an offspring O' such that

$$INST_{O'}(s_i) = \begin{cases} INST_{O1}(s_i) \text{ with probability } 0.5 \\ INST_{O2}(s_i) \text{ with probability } 0.5 \end{cases}$$

Mutation Operator

Our mutation operator selects one segment randomly and assigns it to a randomly determined category. Choose two random integers k1, $1 \leq k1 \leq n$, and k2, $1 \leq k2 \leq m$. Remove s_{k1} from its current category in organism O and attach it to cat_{k2} (i.e., set $INST_O(s_{k1}) = cat_{k2}$).

Fitness Function

For the model, the fitness function is the sum of all satisfied predicates in the semantic net. Let E denote the function. Let $P_{ij}^{(g)}$, be defined as above, with m possible categories. Then

$$E = \sum_{j=1}^{m} \sum_{i=1}^{m} \sum_{g=1}^{r_{ij}} p_{ij}^{(g)} \qquad (1)$$

$P_{ij}^{(g)}$ is a predicate for a relationship between categories, i and j. Each predicate $P_{ij}^{(g)}$ has a corresponding derived predicate, $pred_{ij}^{(g)}(k, l)$, for an analogous relationship between segments s_k and s_l, where s_k is in category i and s_l is in category j. $P_{ij}^{(g)}$ is interpreted based on the

normalised truth value of the derived predicate. Specifically,

$$P_{ij}^{(g)} \quad \begin{cases} \dfrac{\sum_{s_k} \sum_{s_l} \text{Pred}_{ij}^{(g)}(k,\,l)}{|\text{cat}_i| \times |\text{cat}_j|} \\ 0 \qquad \text{otherwise} \end{cases} \tag{2}$$

where $|\text{cat}_i|$ and $|\text{cat}_j|$ are the number of segments classified as category i and category j, respectively. Because all such predicates are not defined between all possible pairs of segments, the normalising factor (the denominator) is subject to redefinition on a case by case basis. Alternatives to (2) are described following the description of derived predicates below.

An example of a fuzzy predicate $P_{ij}^{(g)}$ from our domain is the relationship "is near", where category i "is near" category j. The corresponding derived predicate $\text{pred}_{ij}^{(g)}(k,l)$ describes the relationship between two segments, s_k in category i and s_l in category j. The sum of $\text{pred}_{ij}^{(g)}(k,l)$ for all possible pairs of segments s_k and s_l is normalised by the maximum possible.

Definitions of $\text{pred}_{ij}^{(g)}(k,l)$ are dependent on the underlying semantics of the problem domain. One approach is to define them propositionally as (0,1) if a measurable relationship between s_k and s_l is within or beyond some threshold. A second approach preferred here is to define them as fuzzy truth functions on the interval [0,1]. Inverse exponential truth functions are commonly used in fuzzy set theory to measure the "nearness" of two concepts. An alternative nearness measures are in [ZIE85]. For example, if the description of $P_{ij}^{(g)}$ contains a nominal value (e.g., the SW is approximately 50 kilometers from the NW) then let X_o represent the nominal value and

$$\text{pred}_{ij}^{(g)}(k,l) = e^{-\beta|X_o - X|} \tag{3}$$

where

- X is the observed value corresponding to the same measure (distance, curvature, angle of declination) between s_k and s_l
- β is a constant contrast factor in [0,1] which emphasizes the magnitude of the difference between the observed and nominal value when increased

There are many situations for which the nearness measure is not bounded by an ideal but the closer to s_k the better. In such cases, X_o can be replaced by zero in formula (3).

"Not near" or "as distant as possible" may be measured by the fuzzy complement of (3).

$$\text{pred}_{ij}^{(g)}(k,l) = 1 - f() \tag{4}$$

where f() is the right side of formula (3).

Some relationships such as "above" or "smaller" are not easily modelled as nearness measures. Such relationships can be considered as ordinary propositional truth values.

$$\text{pred}_{ij}^{(g)}(k,l) = \begin{cases} 1 & \text{if } s_k \text{ and } s_l \text{ are so related} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

If there is a measure X associated with the relationship and $X_k > X_l$ when the condition is met, the derived predicate of formula (5) can be represented by the ceiling function

$$\text{pred}_{ij}^{(g)} = \lceil (X_k - X_l)/(|X_k - X_l| + 1) \rceil \tag{6}$$

For $P^{(g)}{}_{ij}$, each object attached to cat_i requires $|\text{cat}_j|$ evaluations of $\text{pred}^{(g)}{}_{ij}$. The multiple evaluations are combined to a single value using fuzzy OR

$$\max_{s_l} [\text{pred}_{ij}^{(g)}(k,l)]; \text{ for each } s_k \text{ in } \text{cat}_i \tag{7}$$

This corresponds to finding the best segment, s_l, that matches the relationship for a given segment s_k. By contract, the combination rule

$$\min_{s_l} [\text{pred}_{ij}^{(g)}(k,l)]; \text{ for each } s_k \text{ in } \text{cat}_i \tag{8}$$

corresponds to fuzzy AND. The heuristic implied by the formula (2) is

$$\sum_{s_l} \text{pred}_{ij}^{(g)}(k,l)/|\text{cat}_j|; \text{ for each } s_k \text{ in } \text{cat}_i \tag{9}$$

which corresponds to the average truth functional value of s_k with all s_l segments in cat_j.

Let $f^{(g)}{}_{ij}(k)$ stand for the segment level combination rule, (7), (8), or (9). Possible aggregation rules to compute $P_{ij}^{(g)}$ are

$$\sum_{s_k} f_{ij}^{(g)}(k)/|\text{cat}_i| \tag{10}$$

$$\max_{s_k} [f_{ij}^{(g)}(k)] \tag{11}$$

$$\min_{s_k} [f_{ij}^{(g)}(k)] \tag{12}$$

which correspond to average, best, and worst match, respectively. The aggregation rule of formula (10) is the one implied by formula (2).

EXAMPLE

Fig. 3 is a reproduction of Fig. 1(a) with most segments labelled (correctly). Eight segments are labelled as s_1, s_2, ..., s_8 and are used below in an example. Table 1 lists and defines all predicates and derived predicates required for the semantic net of Fig. 2. The notation $|\text{cat}_h|$

Table 1. Predicate Descriptions

Predicate	Functional [Pred(h,1)]/normalizer	Description
$p^{(1)}_{co,co}$	$\max_x [\exp(-0.5\,x)]/[\text{coor}]$	near known CS coordinates (distance = x)
$p^{(2)}_{co,co}$	$\max_{x \text{ where } h=1} [\exp(-0.5\,x)]/[(\text{cot}_{co}]-1)$	near other CS segment (distance = x)
$p^{(1)}_{wb,wb}$	$(1/\text{cot}_{wb})\int_x [\exp(-0.5(100-x))]/\text{cot}_{wb}]$	WB diameter near 100 km (distance = x)
$p^{(2)}_{wb,wb}$	$\max_{x \text{ where } h=1} [\exp(-0.5\,x)]/[(\text{cot}_{wb}]-1)$	near other WB segment (distance = x)
$p^{(1)}_{wb,sw}$	$(1/\text{cot}_{sw})\int_i [(x_h-x_1)/([x_h-x_1]+1)\,]/\text{cot}_{sw}]$	WB north of SW (x_h and x_1 are latitudes)
$p^{(1)}_{sw,sw}$	$\max_{x \text{ where } h=1} [\exp(-0.5\,x)]/[\text{cot}_{sw}]-1)$	near other SW segment (distance = x)
$p^{(1)}_{sw,sw}$	$(1/\text{cot}_{sw})\int_x [\exp(-0.5(50-x))]/\text{cot}_{sw}]$	SW 50km from SW (distance = x)
$p^{(2)}_{sw,sw}$	$(1/\text{cot}_{sw})\int_i [(x_h-x_1)/([x_h-x_1]+1)\,]/\text{cot}_{sw}]$	SW north of SW (x_h and x_1 are latitudes)
$p^{(1)}_{sw,sw}$	$\max_{x \text{ where } h=1} [\exp(-0.5\,x)]/[\text{cot}_{sw}]-1)$	near other SW segment (distance = x)
$p^{(1)}_{sw,cs}$	$(1/\text{cot}_{cs})\int_i [(x_h-x_1)/([x_h-x_1]+1)\,]/\text{cot}_{sw}]$	SW north of CS (x_h and x_1 are latitudes)
$p^{(1)}_{cs,cs}$	$(1/\text{cot}_{cs})\int_x [\exp(-0.5(100-x))]/\text{cot}_{cs}]$	CS diameter near 100 km (distance = x)
$p^{(2)}_{cs,cs}$	$\max_{x \text{ where } h=1} [\exp(-0.5\,x)]/[(\text{cot}_{cs}]-1)$	near other CS segment (distance = x)
$p^{(1)}_{o,o}$	$\max_{x \text{ where } h=1} [\exp(-0.5\,x)]/[(\text{cot}_{o}]-1)$	near other O segment (distance = x)
$p^{(*)}_{o,*}$	$(1/\text{cot}_{o})\int_x [1-\exp(-0.5x)]/\text{cot}_{o}]$	not near CS, WB, CB, SW, or SW

refers to the number of segments that are an instance of category h. The value 0.5 is chosen arbitrarily for β in all derived predicates. The exponential form of derived predicates is used for all relationships except "north of" where formula (6) is substituted. The default value for any predicate or derived predicate is zero should a denominator evaluate to zero.

The eight segments distinguished in Fig. 3 are characterized in Table 2. For this example, we need only the geophysical coordinates, the distances between segment centroids, and the distances between the closest points of segments. A larger, more complete description might also contain the length and degree of curvature of each segment.

Table 3 lists six organisms together with their fitness function values which are computed using the predicates in Table 1. The fitness function is given by formula (2). The combination and aggregation rules are formulas (7) and (12), respectively. Derived predicates are variations of formulas (3) and (4) except "north of", which is represented by formula (6) with the requisite measure being latitude. Organism O_1 has no segments labelled incorrectly. O_2 has two segments labelled incorrectly. O_3 through O_6 have 3, 3, 5, and 8 incorrectly labelled segments, respectively. The fitness function values correspond roughly to the correctness of the labelling. Additional predicates (i.e., a more complex semantic net) would improve upon the ordering and separation in most cases.

CONCLUSION

A model for labelling complex scenes via genetic algorithms with fuzzy fitness functions evaluated over semantic nets and GAs is possible. Truth functionals indicating the degree to which specific interfeature relationships are fulfilled are combined at the segment level then aggregated at the category level using fuzzy set operators.

We are currently investigating such issues as the effect of many predicates clustered on one or two



Figure 3. Segmented Image With Correct Labels

Table 2. Segment Descriptors

a. Centroid Position in Fractions of Latitude and Longitude

Segment	Latitude	Longitude
S_1	39.48	70.06
S_2	38.82	68.69
S_3	39.52	66.04
S_4	38.37	66.67
S_5	37.33	66.72
S_6	37.32	66.06
S_7	38.67	65.01
S_8	39.54	64.06

b. Distances Between Centroids (kilometers)

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
S_1	0.00	127.90	257.55	293.03	342.93	375.45	368.60	416.76
S_2	127.90	0.00	164.00	168.39	217.12	247.96	243.55	316.39
S_3	257.55	164.00	0.00	113.01	210.21	209.60	167.14	159.20
S_4	293.03	168.39	113.01	0.00	104.00	90.13	75.67	106.73
S_5	342.93	217.12	210.21	104.00	0.00	96.36	104.29	206.04
S_6	375.45	247.96	209.60	90.13	96.36	0.00	58.67	223.06
S_7	368.60	243.55	167.14	75.67	104.29	58.67	0.00	165.51
S_8	416.76	316.39	159.20	106.73	206.04	223.06	165.51	0.00

c. Closest Proximities (kilometers)

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
S_1	0.00	127.13	-	-	-	-	-	-
S_2	127.13	0.00	-	80.42	-	-	-	-
S_3	-	-	0.00	31.26	-	-	42.00	-
S_4	-	80.42	31.26	0.00	12.72	15.39	16.93	35.00
S_5	-	-	-	12.72	0.00	0.00	-	-
S_6	-	-	-	15.39	0.00	0.00	-	-
S_7	-	-	42.00	16.93	-	-	0.00	20.92
S_8	-	-	-	35.00	-	-	20.92	0.00

Table 3. Fitness Function Values for Selected Organisms

o_1 = <[illegible]> ; $E(o_1)$ = 2.2098

o_2 = <[illegible]> ; $E(o_2)$ = 2.2511

o_3 = <[illegible]> ; $E(o_3)$ = 2.1251

o_4 = <[illegible]> ; $E(o_4)$ = 1.4731

o_5 = <[illegible]> ; $E(o_5)$ = 1.6757

o_6 = <[illegible]> ; $E(o_6)$ = 0.9235

categories, alternate forms for the truth functionals themselves, and the crossover rules. Our image set consists of six segmented infrared photographs of the North Atlantic, each photograph having a different degree of observation. Our testbed will consist of a GA algorithm capable of manipulating the alleles' correspondence to the semantic net.

REFERENCES

[BO87] Booker, Lashon, "Improving Search in Genetic Algorithms", Genetic Algorithms and Simulated Annealing, Lawrence Davis, Ed., Morgan Kaufmann, Los Altos, CA, 1987, pp. 61-73.

[GO86] Goldberg, David E., "A Tale of Two Problems: Broad and Efficient Optimization Using Genetic Algorithms," Proc. of the Summer Computer Simulation Conference, July 28-30, 1986, Reno, Nevada.

[GO88] Goldberg, David E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1988.

[HO75] Holland, John H., Adaption in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michagan, 1975.

[LC87] Lybanon, M. and R.L. Crout, "The NORDA GEOSAT Ocean Applications Program", John Hopkins APL Technical Digest, Vol 8, No. 2, April/June 1987, pp. 212-218.

[RI83] Richardson, P.L., "Gulf Stream Rings", Eddies and Marine Science, A.R. Robinson (ed.), Springer Verlag, New York, 1983, pp. 19-45.

[TB86] Thomason, Michael G. and Richard E. Blake, "Development of An Expert System for Interpretation of Oceanographic Images", NORDA Report 148, June 1986.

[YA75] Yager, Ronald R. "Decision Making with Fuzzy Sets", Decision Sciences, Vol 6, 3, July 1975, pp. 590-600.

[ZA88] Zadeh, Lotfi A., "Fuzzy Logic" Computer, Vol 21, 4, April 1988, pp. 83-93.

[ZI85] Zimmermann, Hans -J., Fuzzy Set Theory and Its Applications, Kluwer Nijhoff Publishing, Dordrecht, The Netherlands, 1985.

EXTENSIONS OF ALGEBRAIC IMAGE OPERATORS: AN APPROACH TO MODEL-BASED VISION[1]

Bao-Ting Lerner[2]
Associate Professor
US Naval Academy
Annapolis, Maryland 21402
and
NASA/Goddard Space Flight Center
Robotics Laboratory
Greenbelt, Maryland 20771

Michael V. Morelli
Electrical Engineering Department
Fairleigh Dickinson University
Teaneck, New Jersey 07666
and
Jackson and Tull Chartered Engineers
NASA/Goddard Space Flight Center
Robotics Laboratory
Greenbelt, Maryland 20771

ABSTRACT

We extend our previous research on a highly structured and compact algebraic representation of grey-level images which can be viewed as fuzzy sets. Addition and multiplication are defined for the set of all grey-level images, which can then be described as polynomials of two variables.

Utilizing this new algebraic structure, we have devised an innovative, efficient edge detection scheme. An accurate method for deriving gradient component information from this edge detector is presented.

Based upon this new edge detection system we have developed a robust method for linear feature extraction by combining the techniques of a Hough transform and a line follower. The major advantage of this feature extractor is its general, object-independent nature. Target attributes, such as line segment lengths, intersections, angles of intersection, and endpoints are derived by the feature extraction algorithm and employed during model matching.

The algebraic operators are global operations which are easily reconfigured to operate on any size or shape region. This provides a natural platform from which to pursue dynamic scene analysis. We discuss a method for optimizing the linear feature extractor which capitalizes on the spatially reconfigurable nature of the edge detector/gradient component operator.

---

[1] This research was supported by the National Aeronautics and Space Administration (NASA) under grant S-92580-D.

[2] Principal Author, to whom correspondence should be addressed.

INTRODUCTION

There have been various algebraic approaches proposed for the description and generation of binary images[2][6]. In this paper we extend our previous work[1][3] on a polynomial architecture based on fuzzy sets for the represention, interpretation and manipulation of grey-scale imagery. An edge detector and a "compass" gradient component operator are defined for this algebraic system, and are employed as the front end of a robust linear feature extractor.

ALGEBRAIC REPRESENTAION OF GREY-LEVEL IMAGES

In a previous paper [1][3] we introduced a polynomial architecture for the description of grey-level (and binary) images. If Z denotes the integers, then an image A may be represented as

$$A = \{(x,y) \in Z \times Z \mid \mu_A(x,y) \in [0,1]\},$$

where $\mu_A(x,y)$, denoting the grey-level of pixel (x,y), is a real number between 0 and 1 with the larger numbers corresponding to the darker grey-levels. In this way A may be viewed as a fuzzy set.

Image Addition and Multiplication:

Let A and B be grey-level images. We define the addition of A and B as:

$$A + B = \{(x,y) \in A \cup B \mid \mu_{A+B} = \min[\max(\mu_A(x,y),\mu_B(x,y)),\max(1-\mu_A(x,y),1-\mu_B(x,y))]\} \quad (1)$$

Observe that in the binary case (with μ = either 1 or 0) the addition reduces to the "exclusive-or" addition defined by Agui, et.al. [2]. That is, in binary the set theoretic addition of images A and B is

$$A + B = (A \cup B) \cap (\bar{A} \cup \bar{B}) \quad (2)$$

where $\bar{A}$ and $\bar{B}$ are the inverse images of A and B respectively.

We define the multiplication of grey-level images A and B by:

$$A * B = \left\{ \sum_{\substack{(x_1,y_1) \in A, \\ (x_2,y_2) \in B}} (x_1+x_2, y_1+y_2) \mid \mu_{A*B}(x_1+x_2, y_1+y_2) = \min[\mu_A(x_1,y_1), \mu_B(x_2,y_2)] \right\} \quad (3)$$

where $\sum$ denotes the addition performed by (1).

Note that the operations of + and * are commutative and associative, thus making the set of all grey-level images a commutative semigroup under each of these operations.

Polynomial Representation of Grey-level Images:

Under the addition and multiplication operations defined in the previous section, we may represent an image A as the polynomial

$$A = \sum_{(m,n) \in A} \mu_{mn} x^m y^n \qquad (4)$$

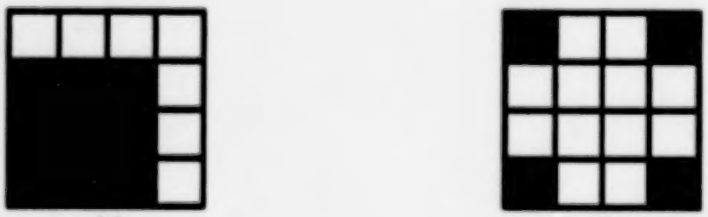
where the coefficient is the grey-level of pixel (m,n) and the summation is defined by (1).

ALGEBRAIC EDGE DETECTORS

Agui, et.al [2] described the classical differential operator

$$D1 = (1+x)(1+y) \qquad (5)$$

and applied it to binary images to produce contour-like figures. The action of D1 on the nine pixel square resulted in a "spotty" edge or contour as depicted in figure 1.



(a) (b)

Figure 1. (a) Simple binary nine pixel image A, (b) the result of D1*A, the differential operator acting on A

We then varied the classical differential operator to the form

$$D2 = 1+xy \qquad (6)$$

This new, more compact irreducible polynomial operator gave a substantially improved contour enhancement capability. Its action on a nine pixel square appears in figure 2.

Although D2 proved to be a simple, efficient operator for contour enhancement in images with a preponderance of vertical and horizontal lines, its major drawback was the introduction of a bias along the -45 degree line, as can be seen in figure 2. In the next section we shall discuss the derivation of accurate gradient component information from analysis of this bias.



(a)

Figure 2. Operation of D2*A.



(b)

Figure 3. Operation of D3*A.

To overcome this bias we varied the differential operator on an image A to

$$D3 * A = \{(m,n) \in (A+Axy) \cup (Ax+Ay) \mid \mu_{D3*A}(m,n) = \max[\mu_{A+Axy}(m,n), \mu_{Ax+Ay}(m,n)]\} \quad (7)$$

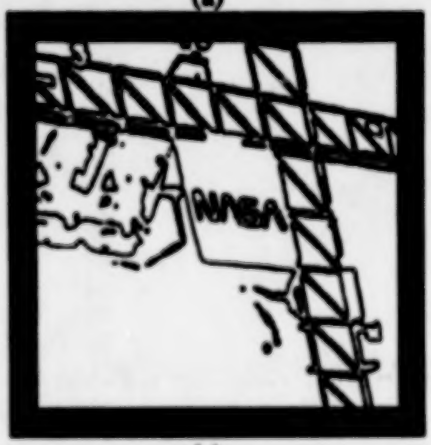
Figure 3 depicts the action of D3 on a nine pixel square. Notice the continuous, unbroken contour. The definition of the min-max addition produces a natural threshold value of 0.5. Unlike classical edge detectors which require the determination of the optimal threshold from the statistics of the contoured image, this natural threshold is a constant for all conditions, thus greatly simplifying the production of the edge from the contour. The edge produced by the action of each of the three defined differential operators on a complicated scene is shown in figure 4a - 4c.



(a)



(b)



(c)

Figure 4. (a) The operation of D1 on a complex scene. Notice the "spotty" edge. (b) D2 applied to the same scene. This edge is better; however the bias is apparent on the diagonal struts. (c) The operation of D3 on the scene. This operator contains no bias and produces a solid edge.

GRADIENT COMPONENT INFORMATION FROM ALGEBRAIC DIFFERENTIAL OPERATORS

We have derived accurate gradient component information by utilizing the bias (mentioned in previous section) resulting from individual differential operations and the underlying algebraic structure. We interpret the differential operation as being a measure of a feature's spatial alignment with that of the differential operator orientation (the differential vector). A characteristic of the addition (1) is that as the differential vector converges to the feature orientation, the number of resultant edge elements (edgels) produced by that operator decreases. At the point which the differential operator and feature are aligned, no edgels are produced by the operator. Conversely, the population of edgels increases as the differential vector diverges from the feature orientation (Fig 5). There is a strong quantitative as well as qualitative correspondence between the number of edgels produced by a differential operator and the orientation of the feature relative to that differential vector.

The differential operation is symmetric and unsigned, hence four operators are required to identify all unique (non-redundant) orientations. We define the four component operators as:

$$G_x = (1+x) \tag{8}$$

$$G_y = (1+y) \tag{9}$$

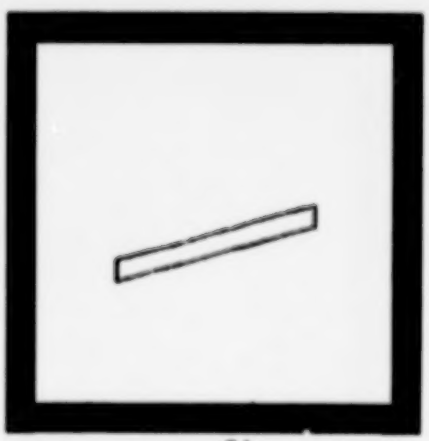
$$G_{xy} = (1+xy) \tag{10}$$

$$G_{xy^{-1}} = (1+xy^{-1}) \tag{11}$$

Notice that the operators (8), (9), and (10) were used previously to define D2 and D3. The operation of each of these component operators on a simple binary test image is shown in figure 5a - 5e.



(a)



(b)



(c)



(d)



(e)

Figure 5 (a) A test pattern oriented 15 degrees relative to the abscissa. This was generated by a Bresenham algorithm. (b) The operation of Gx, (c) Gy, (d) Gxy, (e) Gxy^{-1}. The results of the edgel count for each operation:

Gx : 148
Gy : 516
Gxy : 390

Gxy^{-1} : 662

The image is operated on by each of the gradient operators and the number of edgels contributed by each component is compiled. The resultant component operator with the fewest contributions is the closest to the feature orientation, and is identified as the primary component (Gp). The component operator which contributes the next to the least edgels is identified as the secondary component (Gs). Gp and Gs bound the sector in which the feature is oriented. To determine the actual orientation of the feature relative to the primary component, the inverse trigonometric relationship (12) between the edgel count of Gp and the edgel count of its orthogonal component operator (Gp⊥) is employed.

$$\text{orientation} = \arctan(Gp \text{ count}/Gp\perp \text{ count}) \qquad (12)$$

As an example, consider the test image data of figure 5. The test line is draw with a Bresenham algorithm at an orientation of 15 degrees from the horizontal axis. From the data listed in figure 5 we identify : Gp = Gx, Gs = Gxy and Gp⊥ = Gy. The relative measure from the Gp by (12) is 16.0 degrees (It should be pointed out that accuracy of this operation degrades at very low spatial resolution.)

Notice that (12) generates no sign information, only a relative measure from the primary component vector. The correct placement of this measure from the primary vector (one side or the other) is fixed by the secondary component. Hence letting the global orientation of Gx be zero degrees, the global orientation of the feature in figure 5 is (0+orientation)=16.0 degrees.

A major advantage of this edge detector/gradient component system is that it inherently is a global operation; however reconfiguring it as a spatially symmetric or asymmetric local operation is a trivial matter. This property allows for the compilation of the gradient components over any size and shape regions. In the last section we shall discuss how this reconfiguration is perfectly suited for dynamic scene processing.

LINEAR FEATURE EXTRACTOR

The algebraic edge detector/gradient component operator has been employed as the front end of a linear feature extraction system which is based on the Hough transform. Our primary goal is to establish and maintain object independence and system invariance to object rotation, translation and scaling in the image. The system takes as input the edge and gradient component information (as discussed in the next section) and produces a list of line segments.

The Hough transform is a technique for parameterizing binary image points into a family of curves representing the analytical equations of image features [4][5]. We employed a linear transform which is based upon the form

$$\rho = x^{*}\cos(\theta)+y^{*}\sin(\theta), \qquad (13)$$

where x and y are the coordinates of the edgel, θ is the evaluated orientation relative to the x axis (evaluated in the range of $-\pi/2$ to $\pi/2$), and ρ is the normal vector to the line of that orientation. The evaluation of (13) produces the transform parameter space.

We have developed an adaptable and reliable method for processing the Hough transform parameter space to extract unique image feature equations[3]. The analytical feature representations are then utilized to locate actual feature segment endpoints via a line following/segment locating mechanism.

The processing sequence of the linear feature extraction system is shown in figure 6a - 6e. Details of this system can be referenced in [3].



(a)



(b)



(c)



(d)



(e)

Figure 6. (a) The operation of D3 on an ORU handle scene. (b) Hough parameter space based on the normal equation of a line. Rho is the ordinate, theta the abscissa. (c) Peak detected parameter space. (d) The analytically derived lines superimposed on the edged image. Note the accuracy. (e) The line segments determined by the line follower/line locator. This infomation is passed to a model matcher.

DYNAMIC SCENE ANALYSIS

The major drawback of the Hough transform is the large amount of computation that it requires. It can be seen from (13) that the number of iterations will equal the number of edgels multiplied by the theta resolution. It is apparent that the optimization of the transform requires the reduction of the parameter space theta resolution as well as the area of the image which is transformed. The intent is to perform this optimization in a way which does not violate the goals of object independence and system invariance to rotation, translation and scaling. This can be accomplished by evaluating the image only in desired areas and at high theta resolution at selected orientations. This optimizing information can be derived, in a dynamic sence, from previous processing. In previous work [3] we discuss the optimization of the Hough parameter space via the utilization of information from a geometric world model.

We have recently developed a mutually optimizing system in which spatial and gradient information from the algebraic edge/gradient component operator optimizes the Hough transform, and the confirmed object feature information is used to redefine the size and shape of the algebraic operator. In this way the highest degree of accuracy, optimization and confirmation is achieved.

The unique ability to redefine the size and shape of the algebraic edge detector/gradient component operator provides a natural environment to perform this optimization.

CONCLUSION

The algebraic approach provides an innovative methodology as well as a sound mathematical framework for image representation and manipulation. It is significant that this algebraic system provides insight into the morphological structure of the image itself, as well as the operators on the image. Real-time implementation of these operations can be easily realized due to the efficient min-max nature of our algebraic system. The simple extraction of the gradient component information from the polynomial structure demonstrates the strong potential of this approach.

ACKNOWLEDGEMENTS

The authors would like to thank David Provost, Danny Dalton, and Stanford Ollendorf of NASA/Goddard Space Flight Center Flight Telerobotics Laboratory for their encouragement and support of this research. This research was primarily funded under NASA grant S-92580-D. The original research leading to this paper was performed by Professor Lerner under NSF grant IST-S409621.

REFERENCES

[1] B. Lerner and H. Thomas, "Polynomial Representation of Grey-level Images", SPIE Applications of Artificial Intelligence VI, Vol. 937, April 1988, pp. 104-116.

[2] T. Agui, M. Nakajima, and Y. Arai, "An Algebraic Approach to the Generation and Description of Binary Pictures", IEEE Trans.on PAMI, Vol. PAMI-4, No. 6, Nov. 1982, pp. 635-641.

[3] B. Lerner, M. Morelli, and H. Thomas, "Applications of Algebraic Image Operators to Model-Based Vision", SPIE Applications of Artificial Intelligence VII, Vol. 1095, March 1989, pp. 36-47.

[4] D. H. Ballard and C. M. Brown, *Computer Vision*, Englewood Cliffs, N.J. : Prentice-Hall 1982.

[5] J. Illingworth and J. Kittler, "A Survey of the Hough Transform", Computer Vision, Graphics, and Image Processing, No 44, 1988, pp. 87-116.

[6] G.X. Ritter and P.D. Gader, "Image Algebra Techniques for Parallel Image Processing", Journal of Parallel and Distributive Computing 4(5), March 1987.

FUZZY-C DEVELOPMENT SYSTEM: A COMPLETE OVERVIEW

C. Perkins, J. Teichrow, and E. Horstkotte
Togai InfraLogic, Inc.

(Paper not provided by publication date.)

AUTHOR INDEX

Adam, Susan C. 229
Akin, D. 49
Allen, J. 381
Almadjian, M. 255
Andary, J. 311
Ankenbrandt, C.A. 679
Atkinson, David J. 281
Aume, Nilss M. 139
Baffes, Paul 51,525
Bailey, Robert W. 447
Barhen, J. 523
Becker, A. 337
Bejczy, A.K. 429,649
Bertrand, W.T. 257
Biegal, John E. 579
Bonadies, Gregory A. 353
Brill, John 611
Brody, Adam R. 471
Buckles, B.P. 679
Bull, John 113
Burke, Thomas E. 415
Cheeseman, P. 543
Christiansen, E. 521
Chutjian, A. 263
Cleghorn, Timothy F. 447
Colombano, Silvano P. 187
Comstock, J. Raymond, Jr. 353
Cooke, D. 169
Cross, Jon B. 269
Davis, R. 667
Dee, Jurgen 477
Dember, W. 337
Dewberry, Brandon S. 195
Dingler, J. 279
Dionise, Joe 551
Dobryden, Al 551
Dominick, Jeffrey 113
Donahoo, M.E. 99
Donnelly, J. 95
Duffie, Neil A. 637
Edwards, G. 405
Ehlers, H.K.F. 171
Eike, David R. 585
Elgin, J. 169
Ellis, Stephen R. 465
Erickson, J.D. 559
Fairchild, K. 669
Falco, P.M. 257
Ferguson, Dale C. 509
Firby, R. James 423
Fowler, M.E. 99
Galinsky, T. 337
Garrett, Henry B. 361,493
Gat, Erann 423
Georgeff, Michael P. 209
Gersh, Mark A. 13
Gillam, April 219
Gillan, Douglas J. 223,661
Gluckman, J. 337
Gnabasik, M. 279
Green, B. 255
Gregory, J. 277
Groleau, Nicolas 187
Guidice, Donald A. 515
Hacisalihzade, Selim 465
Halterman, K. 311
Hammen, David G. 295
Hancock, P. 337
Hankins, W. III 647
Hanson, Matthew A. 545
Harris, Randall L., Sr. 353
Hartley, Craig S. 671
Haymann-Haber, Guido 187
Healey, Kathleen 113,559
Hedgecock, J. 49
Heher, Dennis 117
Hess, Clifford W. 441
Hewitt, D. 311
Hoffman, R. 167
Holt, J.D. 257
Horstkotte, E. 697
Hughes, Peter M. 289
Humphrey, Darryl 339
Iliff, Richard 133
Ingrand, Francois Felix 209
James, Jeffrey M. 347
James, Mark L. 281
Jaster, M. 645
Jenkins, J. 251
Jensen, Dean G. 229
Kan, Edwin P. 567
Kiech, E.L. 257
Kim, W.S. 649
Kiss, Peter A. 605
Kittle, Frank P., Jr. 299
Kofsky, I.L. 159
Korna, M. 137
Kosmo, Ioseph 61
Kramer, Arthur 339
Krawczonek, Walter M. 181

Lerner, Bao-Ting 687
Lewis, Robert 661
Li, Larry C. 441
Liu, Andrew 477
Lollar, Louis F. 107
Lybanon, M. 679
Maag, C. 179
Maley, P.D. 99
Malone, Thomas B. 485
Maris, M.A. 159
Matijevic, J.R. 321
McConnell, J. 483
McDaniel, Joe W. 143
McKee, Sandra D. 149
MecKlinger, Axel 339
Mende, S. 105
Meredith, G. 669
Miller, David P. 423
Mitchell, Tom 409
Mixon, R. 647
Montemerlo, M. 131
Moore, J. 319
Morelli, Michael V. 687
Morgan, T. 93
Mount, Frances E. 149
Munro, Paul 305
Murad, Edmond ... 97,159,169,263
Muratore, J. 279
Murphy, G.B. 493
Myers, Robert M. 295
Odom, James B. 1
Ohm, T. 429
Orient, O.J. 263
Paddock, Eddie J. 299
Parker, Alice C. 219
Parker, Ellen M. 593
Perkins, C. 697
Permenter, Kathryn E. 485
Pervaiz, M. 169
Petry F.E. 679
Phinney, D.E. 559
Pike, C. 255
Pocklington, Tony 299
Pownall, Paul 117
Probe, John D. 155
Pugh, George E. 415
Quinn, Todd 181
Quiocho, Leslie J. 447
Rasmussen, A. 279
Regian, J. 577
Rienhardt, A. 91,137
Reuter, G.J. 559
Richards, Dale W. 123
Rosenchein, S. 407
Rosenthal, Don 187
Roth, Mary Ellen 181
Rouen, Michael 61
Rudisill, Marianne 661
Sabelhaus, P. 311
Sanderson, Penelope M. 347
Satterwhite, R. Scott 633
Seamster, Thomas L. 585
Seiber, B.L. 257
Seidel, Jorge P. 219
Seidler, Karen S. 347
Simmons, Reid 409
Sirevaag, Erik 339
Sliwa, Nancy E. 537
Smith, Karl U. 235
Smith, Randy L. 223
Smith, Stephen 465
Smith, Thomas J. 235
Snyder, David B. 393
Sorenson, E. 49
Spoon, D. 253
Stark, Lawrence 477
Stevens, N. John 383
Stillwell, R.P. 383
Stock, M. 631
Stramler, James H. 229
Stuart, Mark A. 223
Swanson, G. 105
Szakaly, Z. 429,649
Szolovits, Peter 187
Tautz, M. 169
Taylor, Eric W. 545
Teichrow, J. 697
Tharp, Gregory 477
Thremann, J. 337
Tran, N.H. 159
Trumble, Terry M. 127
Truong, Long V. 181
Truszkowski, Walter 585
Vidulich, Michael A. 329
Villareal, James 525
Walker, Michael W. 551
Walls, Bryan 107
Walters, Jerry L. 181
Wang, Jyhshing Jack 455
Wang, Lui 51,299
Warm, J. 337
West, Phillip 61
Wiker, Steven F. 637
Wilkenson, D. 381
Will, R. 47
Wilmington, Robert P. 229
Wong, Brenda 477
Wood, B.E. 257
Woolley, C. 319
Yamashita, Hitomi 477

Wilmington, Robert P. 229
Wong, Brenda 477
Wood, B.E. 257
Woolley, C. 319
Yamashita, Hitomi 477
Yorchak, John P. 671
Young, Laurence R. 187
Zik, John J. 637
Zweben, M. 407

*U.S. GOVERNMENT PRINTING OFFICE: 1990 - 727-069 80013

NASA
National Aeronautics and Space Administration

REPORT DOCUMENTATION PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CP-3059		

4. Title and Subtitle	5. Report Date
Third Annual Workshop on Space Operations Automation and Robotics (SOAR '89)	March 1990
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Sandy Griffin, Editor	S-599
9. Performing Organization Name and Address	10. Work Unit No.
Lyndon B. Johnson Space Center Houston, Texas 77058	11. Contract or Grant No.
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
National Aeronautics and Space Administration, Washington DC 20546 U.S. Air Force, Washington, DC 23304 University of Houston - Clear Lake, Houston, Texas 77058	Conference Publication
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Papers presented at the Third Annual Workshop on Space Operations Automation and Robotics (SOAR '89), hosted by the NASA Lyndon B. Johnson Space Center at Houston, Texas, on July 25-27, 1989, are documented herein. During the 3 days, approximately 100 technical papers were presented by experts from NASA, the USAF, universities, and technical companies. Also held were panel discussions on Air Force/NASA AI Overview and Expert System Verification and Validation. Tutorial sessions included Neural Networks; Theory and Application of Back Propagation; Verification and Validation of Expert Systems; Evaluation of Expert System Tools; and Technical Environment for Modular Architectures for Robotics in Space; and are not documented herein. Technical topics addressed included intelligent systems, robotics, human factors, and environment.

17. Key Words (Suggested by Author(s))		18. Distribution Statement
Knowledge representation	Expert systems	Unclassified - Unlimited
Teleoperators	Knowledge engineering	
Human factors engineering	Manipulators	
Robotics	End effectors	Subject Category: 59
Artificial intelligence	Spacecraft contamination	

19. Security Classification (of this report)	20. Security Classification (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	720	A99

For sale by the National Technical Information Service, Springfield, VA 22161-2171

JSC Form 1424 (Rev Aug 89) (Ethernet Jan 88)